Relevant Name Search

User Guide

Version 2.0

Table of Contents

Table of Contents 2

RNS Overview 4
- Why Choose RNS? 4
- RNS Customer Portal 5
- RNS Customer Support 5

Core Features 6
- Relevancy 6
- Flexibility 6
  - Full-Search Solution 6
  - Partial-Search Solution 7
- Customization 7
- Availability 7
- Reporting 8

Under the Hood 9
- Domain Rank 10
  - Rank vs Score 10
  - Weights 11
  - Filtering 11
- TLDs: Generic vs Vertical 12
  - Vertical TLDs 12
  - Generic TLDs 12
- RNS Data Points 13

Examples and Use Cases 15
- The Default Query 16
- Increasing the Weight of Generic TLDs 18
- Placement and Presentment 20
- Force TLDs to the Top of Result List 22
- Domain Availability 24
- Advanced UI Filtering 25
  - Showing User-Selected TLDs 25
  - Exact-Match SLDs 26
- Targeting TLDs by User Location 27
  - RNS Customer Portal 28
- Tracking Query Metrics 30
  - Two Queries 30

A/B Testing 31
Premium Domains 33
Default Behavior: Exact SLD Match 34
Return Any Relevant Domain 35
Return None 37
Adult and Offensive Names 38
Supporting Your TLDs 39
Hints: Custom Keyword Categories 40
User Interface 40
Customer Portal 41
API Query 42

RNS Overview

RNS is a domain suggestion tool with one purpose: Empowering users to find that perfect domain name. At Donuts, we believe the path to achieving this is through a focus on *relevant* domain suggestions. How can this relevancy be achieved? It requires leveraging *all* available TLDs combined with generating intelligent and contextual SLD suggestions. These two elements power RNS to recommend uniquely diverse, meaningful, and useful domain suggestions.

It is from this vision that RNS works to achieve the ultimate goal: enabling users to create that perfect domain.

Why Choose RNS?

RNS provides one of the most versatile domain search solutions in the industry. Our service has the most to offer, which includes returning the most relevant domain suggestions, performing live availability checks, and supporting a wide breadth of TLDs.

Free to Use	RNS is completely free! Building your own search engine or using third-party solutions can cost thousands of dollars. Instead, invest that money into other important business activities.
Full-Search Solution	RNS is ready to use immediately out of the box, and integration is simple.
Partial-Search Solution	RNS can be used on top of any existing search engine. Utilize RNS to satisfy specific use cases, such as presentment campaigns. Expand domains offered, such as adding a suggested domain box.
Relevant	Our intelligent recommendation system incorporates contextual signals, historical frequency data, and nearly 1000 TLDs to recommend the most relevant domains for your users.
Domain Availability	Be assured that all domain results are available for registration by utilizing our unique on-demand availability service.
Flexible	RNS is the only third-party search solution that allows you to influence and control search results for your needs. It is the only search solution that can support advanced UI search form features.

RNS Customer Portal

We provide an easy-to-use customer portal, allowing you to configure vital data, manage your account, access our documentation, and contact customer support.

The RNS Customer Portal can be accessed at: https://app.rns.domains/

RNS Customer Support

We are here to help! If you have any questions, clarifications, or concerns, please reach out to our customer service team. We want to ensure that you fully understand what RNS has to offer and how our service can best meet your needs.

Email: rnssupport@donuts.email

Core Features

Relevancy

Relevant domain suggestions are the core of our solution. RNS strives to return domains that are relevant at both the SLD *and* TLD level, providing results that best match user input. We are able to provide this functionality, because we have invested years into research, data analysis, and careful relevancy balancing. Broadly, this includes incorporating the following:

- Generating intelligent SLD suggestions, utilizing historical domain frequency matching, contextual language signals, and other key heuristics.
- Selecting from more than 900 TLDs, choosing the best matches based on user input.
- Appropriately weighting and scoring potential domains, returning only the most relevant recommendations.

Flexibility

RNS can be adapted to satisfy a variety of business use cases. As a full domain search solution, our service is able to recommend domains right out of the box without configuration. Alternatively, you can implement RNS as a partial search solution, configuring out product to solve your specific needs.

For these reasons and more, RNS was made to be as flexible as possible. We provide a number of query options, allowing you to tailor results how you see fit. We invite you to review the Examples and Use Cases section, as it showcases how you can best achieve the most common goals.

In framing how RNS might be used, we like to present two separate concepts:

Full-Search Solution

RNS has the power and capability of being *the* domain search solution for your system. It is simple to support our API service, and you can trust its ability to return the most relevant domain suggestions for your users. Additionally, we provide a number of filtering and data customization options, allowing you to adjust results exactly as you need. This is specifically powerful when you need to ensure that certain results are always display, or when you need to support advanced UI search features.

Partial-Search Solution

RNS can also be used as a partial-search solution. This option allows customers to expand their current search service without making large structural changes. Through the use of filtering parameters, you can tailor results for specific use cases such as diversifying suggested domains or providing relevant results for presentment campaigns.

No matter how you decide to use RNS, our service is always adapting and growing. If RNS can't solve your business needs, then please let us know! We are here to ensure that you can deliver the best search experience for your users.

Customization

The RNS Customer Portal allows you to globally customize settings for all of your queries. These will help tailor RNS results to best meet your needs. Functionality includes:

- Selecting the TLDs that you want to support.
- Manually increasing the score of specific TLDs for generic user searches.
- Choosing TLDs to return based on a user's geographic location.
- Returning contextual TLD recommendations through our *hints* feature.

Once an account has been created for you, the RNS Customer Portal can be accessed at: https://app.rns.domains/

Availability

It is top priority for us to return domain suggestions that are actually available for registration. As a result, we have built a robust system to ensure that RNS can *quickly* return *available* domain suggestions, and we pride ourselves in our accuracy.

Nonetheless, given that availability changes daily, it is impossible to be completely certain that a domain is available without performing an on-demand check. If live availability is your requirement, then RNS offers a special service called *Enhanced Availability*. By making real-time DNS queries for suggested domains, Enhanced Availability provides increased accuracy for domain availability. More information on this parameter is provided under the Examples and Use Cases section.

Reporting

As a customer of RNS, we will provide you with regular reports containing key data metrics. These reports will center on overall account activity, user search characteristics, RNS recommendations, and estimated conversions. It is our desire that these reports will act as a catalyst, allowing you to better understand user behavior and identify opportunities for improving RNS results going forward. We want to ensure that your user search experience as effective as possible!

Under the Hood

We make our best attempt to simplify your interaction with the RNS system. However, the method by which RNS scores and ranks domain suggestions is a complicated process. In order to best understand how to query our API and how to interpret the results provided, we believe that it’s necessary to have a high-level understanding of some RNS concepts. Please feel free to initially scan this section in the beginning. After you start testing and integrating RNS, the content provided here may become clearer.

Domain Rank

When you query RNS, our API will return a list of recommended domains. Each domain in this list is assigned a *rank* value, and the result list order is dependent on this rank. This makes the rank value the most important data point for a given RNS query. Rank is described in more detail below.

Rank vs Score

One of the more important concepts to understand is the difference between the rank and score for a given domain recommendation. As described in the introduction to this section, *rank* is the actual numerical position of a domain in an RNS result set. Rank is entirely dependent on a domain's final score.

The *score* is the final calculated value that RNS gives to a recommended domain. The score of a domain varies based on relevancy, account configuration, query parameters, and internal RNS logic. The score can be a value between 0 and 1.

As an example, let's say one of your users searches for "seattle restaurant". In the table below, you can see how the calculated domain score affects the result rank.

Domain Recommendation	Score	Rank
seattle.restaurant	1.00	1
seattlerestaurant.menu	1.00	2
seattlerestaurant.equipment	0.94	3
seattlerestaurant.catering	0.85	4
washingtonrestaurant.menu	0.80	5

The most important points to know about rank and score are the following:

1. In all cases, you cannot explicitly set the final *rank* position of a recommended domain. In *almost* all cases, you cannot explicitly set the final *score* of a recommended domain
2. You can increase the *score* of a recommended domain by affecting its *weights*. Weights can be influenced in a number of ways, which will be discussed in a following section.
3. You can influence the rank of a result set by filtering out domains that you do not wish returned. *Filtering* is discussed in a following section.

Weights

RNS recommendation logic automatically calculates the score and rank for a domain within a result set. This score is based on a concept called relevancy *weights*. Weights are mechanisms for increasing or decreasing the score of a recommended domain. They can be applied to the SLD, the TLD, or to the entire domain itself. Weights, then, are one of the most important and powerful ingredients of the RNS system. Most weights are protected within the RNS system, but some weights can be customer influenced.

We allow customers to influence a number of domain weights through our API or by configuring account data in the RNS Customer Portal. Most often, customers want to manually increase weights for the SLDs and TLDs most desired by their users. As an example, let’s say that you want to increase the score for specific TLDs based on the location of your user. RNS supports this action through a feature called *geotargeting*. In the context of geotargeting, if one of your users makes a query from Germany, then domain recommendations containing the country code TLD, “de”, will have an increased weight, which leads to a higher calculated score.

For more examples and ideas on how to influence domain weighting, please reference our Examples and Use Cases section.

Filtering

Filtering is the action of removing certain domain recommendations from a result set. This action is normally taken when specific domain qualities are undesired for a given search. The result of using filters is that higher scoring, undesired domains are removed the list of suggested domains, moving lower scoring, desired domains moving up to the top. In effect, filters increase the rank these desired domains.

RNS provides a number of filtering options to help tailor results for your needs. As an example, let’s say that you do not want to display premium domains to your users. By passing a filter parameter, you can remove all premium domains from the returned list of results.

TLDs: Generic vs Vertical

With the addition of new gTLDs, there are now myriad TLD options for a user to choose from. Many of these are positioned to fit a specific purpose, such as "realestate" or "lawyer". RNS conceptualizes these TLDs as *vertical*. Other TLDs are less specific, meaning that you cannot apply a single purpose to them. RNS considers these TLDs as generic. The following sections describe how this difference affects RNS results.

Vertical TLDs

V*ertical* TLDs can be thought of as having a context and association within a given language. For example, the TLD "lawyer" has a clear meaning that can be related to other ideas in the English language, such as law, legal, and attorney. In another example, the TLD "club" might be associated with words such as venue or dance, or, in another context, might be mapped with words such as weapon or tool.

These associations are a key feature of RNS, and they have a profound influence of RNS scoring. If a user's input can be mapped to a TLD, then domain results containing that TLD can be considered *vertically* weighted. In an example, when a user searches for the word "law", they are likely to be recommended a domain for TLDs such as "attorney", "lawyer", and "degree", because they are all vertical matches. Since RNS is a relevancy-focused search solution, if a search is vertically weighted, then it will *not* be generically weighted. Generic TLDs are discussed in the next section.

Generic TLDs

Generic TLDs do not have a context or language association. Consequently, they cannot be connected to any keywords, and vertical weighting for the TLD cannot be applied. For example, the TLD "com" does not naturally relate to a specific idea in the English language, so keywords cannot mapped to it. This also applies to other TLDs such as "net", "org", and "xyz".

In this scenario, RNS primarily relies on domain name frequency data in order to select the most relevant TLDs. This *generic* weighting is accomplished by analyzing historical signals and patterns. As a result, even if the input is not contextual, this method can produce very relevant recommendations.

Additionally, we do recognize the importance of generic TLDs in the domain landscape. To further support these TLDs, we provide a tool that allows customers themselves to manually set generic

TLD weight levels. This gives you more control to influence RNS results according to your needs. The Examples and Use Cases section provides more details on how this can be applied and used.

RNS Data Points

RNS returns a list of recommended domains for a valid search query. This output contains many data points that customers can use for internal data analysis. You can choose how much data you want using a query parameter called *verbose*. Please refer to the RNS API Guide for more information on this subject.

For the purpose of this guide, we will describe each data point and how you can use it for your own tracking purposes.

- **Domain:** The following items are specific to the recommended domain itself:
    - **Rank**: The position of the recommended domain in the returned list. Domains are returned in order of their rank. For example, a rank of 3 means that the domain was returned in the third position in the result set.
    - **Domain:** The name of the recommended domain.
    - **Score:** The calculated score that RNS has assigned to the recommended domain for a specific query. This score is generated based on a number of relevancy factors. The higher the score, the higher the *rank* of the domain.
    - **Exact:** If RNS evaluates that the recommended domain name is an exact match of the user input, then *exact* will be true. For example, if a user searches for “rns.technology”, and the domain, “rns.technology”, is available, this will be considered an exact match.
    - **Registered:** If the recommended domain is currently unavailable for registration, then *registered* will be true. *Note*: This value will always be false, *unless* your query parameter specifically allows registered domains to be returned.
    - **Premium:** If the recommended domain is flagged in our system as a premium domain, then *premium* will be true.
    - **Platinum:** If the recommended domain is flagged in our system as a platinum domain, then *platinum* will be true. Note: Platinum domains are a special tier of domains that are generally priced much higher than premium domains.
    - **Adult:** If RNS considers the recommended domain to contain “mature” words, then *adult* will be true.
    - **Offensive:** If RNS considers the recommended domain to contain offensive words, then *offensive* will be true.
- **SLD:** The following items are specific to only the SLD of the recommended domain:
    - **Name:** The name of the SLD.

    - **Value:** The calculated value that RNS has given to the SLD. This score is generated based on a number of relevancy factors.
    - **Adult:** If RNS considers the SLD to contain “mature” words, then *adult* will be true.
    - **Offensive:** If RNS considers the SLD to contain offensive words, then *offensive* will be true.
- **TLD:** The following items are specific to only the TLD of the recommended domain:
    - **Name:** The name of the TLD.
    - **Value:** The calculated value that RNS has given to the TLD. This score is generated based on a number of relevancy factors.
    - **Lucene:** If RNS matches the user input to a TLD based on historical frequency data, then *lucene* will be true.
    - **Manual:** For a given query, if RNS has recommended a TLD because it was provided for specific query parameters, then *manual* will be true. For example, if *onlyTLDs=com* is provided in the query, then all “com” recommended domains will have a *manual* value of true.
    - **Category:** If RNS matches the TLD based on relevant keyword categories, then *category* will be true. Essentially, this means that keywords in the SLD are mapped to a TLD based on relevancy matching.
    - **Geo:** If the TLD is returned because of geographic targeting, then *geo* will be true.
    - **Graph:** *Graph* will be true if the recommended TLD is relevancy-mapped to the TLD provided in the user input.

Examples and Use Cases

The goal of this section is to illustrate how you can use RNS to meet your business needs. Our service can be utilized in a number of different ways, and we want you to feel comfortable that your RNS implementation is always returning the best suggestions to your users.

In the examples below, we've attempted to capture the most common and relevant use cases. We invite you to use these as a template for your needs. Additionally, please use our accompanying RNS API Guide for specific implementation details. If you have any questions, or if you have any problems getting things exactly as you desire, please contact us!

The Default Query

Parameters Discussed: *account, password, input, maxCount*

RNS is simple to use right out of the box. We want you suggesting relevant domains with as little setup as possible. In order to perform a basic query, you only need to provide the following parameters: *account*, *password*, *input,* and (optionally) *maxCount*. Your *account* and *password* values will be provided by our customer service team. Additionally, you may access them via the RNS Customer Portal (Account Management > Account Details).

In our first example, the RNS customer, *Domains 4 All*, has already been allocated an account *password*. In order to test RNS, they would simply send the following query.

Search Query:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&
input=rnssearch
```

Search Results:

Rank	Suggestion
1	rnssearch.rocks
2	rnssearch.live
3	rnssearch.site
4	rnssearch.social
5	rnssearch.online
6	rnssearch.xyz
7	rnssearch.news
8	rnssearch.video
9	rnssearch.pics
10	rnssearch.buzz

Note that RNS returns 10 results by default. If you need to return a different amount, you can use the *maxCount* parameter. *Domains 4 All* displays up to 25 suggested domains on their results page, so they would send the following query:

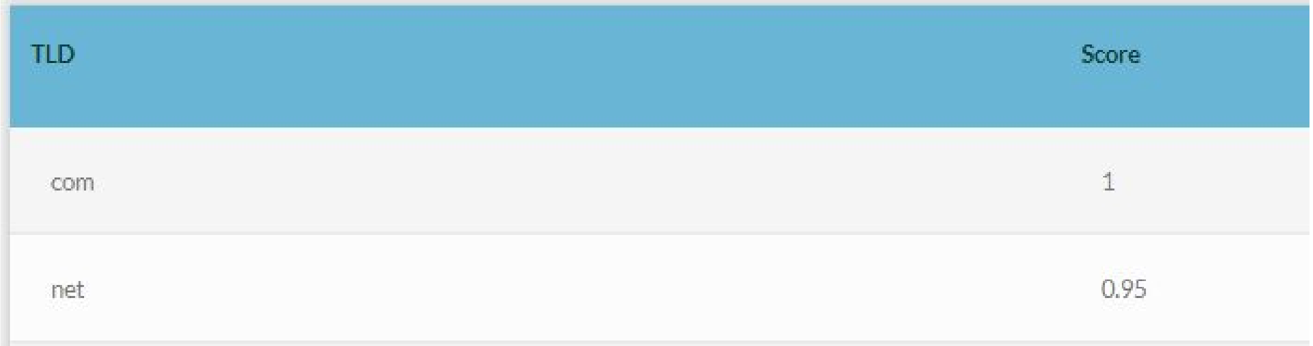
Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=rnssearch&**maxCount**=25

Increasing the Weight of Generic TLDs

In the Under the Hood section, we described the difference between Generic TLDs and Vertical TLDs, and we explained the difficulty in fairly scoring generic user inputs for relevancy. While RNS does its best to provide good recommendations in these scenarios, we want to allow customers the ability to manipulate generic results as well. In the example below, we will show you how you can increase TLD weights specifically when a user input is identified as *generic.*

In the RNS Customer Portal, we provide a page for configuring the weight levels for generic TLDs, and these weights will apply whenever the user provides a generic input. All you need to do is select a TLD and provide a “score” value, ranging from 0 to 1. The higher the value, the larger the increase in weight for the TLD. In the screenshot below, we provide an example weight for two generic TLDs.



TLD	Score
com	1
net	0.95

Please note: It may take up to 20 minutes for the RNS API to reflect changes made in the Customer Portal.

Reusing the same query from The Default Query section, you can easily see the difference in results after making this generic TLD change.

Search Results

Rank	Suggestion
1	**rnssearch.com**
2	**rnssearch.net**
3	rnssearch.rocks
4	rnssearch.live

5	rnssearch.site
6	rnssearch.social
7	rnssearch.online
8	rnssearch.xyz
9	rnssearch.news
10	rnssearch.video

Placement and Presentment

Parameters Discussed: *onlyTLDs, maxCount*

As discussed in the *Flexibility* section of this document, RNS can be used as a partial-search solution for a number of use cases. One of the most powerful examples of this is supporting placement and presentment needs. Often domain sellers will want to show specific TLDs in a distinct location on their web page.

As an example, *Domains 4 All* may want to show recommendations only for the TLD "live” at Position 2 in their search results. In this scenario, *Domains 4 All* can make a single specific query to RNS in order to retrieve the most relevant result for "live” by using the *onlyTLDs* parameter. In addition to this, they may choose to set the *maxCount* value to 1, since only one suggestion is needed.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=musicevents&**onlyTLDs**=live&**maxCount**=1

Search Results:

Rank	Suggestion
1	musicevents.live

In a second scenario, let’s say that *Domains 4 All* has signed a promotional agreement with a Registry. The agreement states that *Domains 4 All* must display the most relevant result in Position 2 for a selected group of TLDs. In the RNS query, the *onlyTLDs* value would be modified to contain each TLD in this group.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=musicevents&**onlyTLDs**=bingo,casino,dating,dog,events,movie,show,singles,theater,actor,dance,live,band&**maxCount**=1

Search Results:

Rank	Suggestion
1	music.events

Force TLDs to the Top of Result List

Parameters Discussed: *forceTLDs*

Customers may desire to use RNS as a full-search solution, but they also need to have control over the results displayed in top positions. Specifically, they want to ensure that certain TLDs always appear at the top of the results page, and display all relevant results below that. RNS supports this need by providing a parameter called *forceTLDs*.

In order to use *forceTLDs*, you must provide a TLD and an associated score. RNS guarantees that a recommended domain will be returned for that TLD at the specified score. For example, let's say that *Domains 4 All* is using RNS as a full-search solution, but they want to ensure that TLDs "com" and "org" appear at the top of the results list. They may achieve this by supplying *forceTLDs* in the format below.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=rnssearch&**forceTLDs**=com:1.0,org:0.9

Search Results:

Rank	Suggestion
1	**rnssearch.com**
2	**rnssearch.org**
3	rnssearch.rocks
4	rnssearch.live
5	rnssearch.site
6	rnssearch.social
7	rnssearch.online
8	rnssearch.xyz
9	rnssearch.news
10	rnssearch.video

Important Note: The *forceTLDs* parameter does *not* guarantee rank. It only guarantees the score of the domain. If you provide a score of 1.0, it is still possible for a domain recommendation to appear *above* the forceTLD result, because it has a higher rank. As an example of this, a domain that is an exact match of the user input will always appear at the top of the list.

Domain Availability

Parameters Discussed: *enhancedAvailability*

It is very important for customer search results to contain domains that are available for registration. If a user selects a registered domain, and that user is unable to complete their purchase at checkout, then this results in a frustrating user experience.

In this light, domain availability is a high priority for us, and we work hard to ensure that RNS returns results both quick *and* accurate. On a daily basis, our system compiles the latest set of registered domains, ensuring that they will not be returned for recommendation. Unfortunately, due to the variability of domain information, we cannot guarantee 100% coverage of all registered domains for our default queries.

RNS does provide a very unique and capable alternative option called Enhanced Availability. When the *enhancedAvailability* parameter is enabled for a query, RNS will perform live availability checks for all recommended domains. This ensures that each returned domain is 100% available for registration. As our customer, this can give you peace of mind that RNS will never return a registered domain.

Important Note: We must provide a warning before using this feature. Making live availability checks increases the latency for an RNS API call. This latency increase can vary depending on the number of domains requested and the frequency of your requests. We ask that you thoroughly test Enhanced Availability to make sure that these increases are acceptable for your needs.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=rnssearch&**enhancedAvailability**=1

Advanced UI Filtering

Parameters Discussed: *onlyTLDs, exactSldOnly*

Each customer provides a unique domain search form for their users. Within this form, customers may choose to give their users the freedom to filter and tailor results specifically for their needs. In order to accomplish this, they may provide advanced filtering options next to the search field. In support of these UI features, RNS supplies a number of query parameters for filtering results.

To exemplify UI filtering, *Domains 4 All* wants to provide two user options:

1. Show results only for user-selected TLDs
2. Only display exact-match results for the provided SLD.

The examples below illustrate how they might accomplish this.

Showing User-Selected TLDs

A *Domains 4 All* user wants to create a travel blog. They don't know exactly which domain they will choose, but they have some ideas in mind. Using the search filter tool, they specifically select the TLDs that they want to see, filtering out all others. In our example, the user chooses the TLDs, "blog" and "travel", and provides the search input, "newvistas".

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=newvistas&**onlyTLDs**=blog,travel

Search Results:

Rank	Suggestion
1	newvistas.travel
2	newvistas.blog
3	new-vistas.travel
4	new-vistas.blog
5	newvista.travel

6	latevistas.travel
7	newvista.blog
8	new-vista.travel
9	mynewvistas.travel
10	modernvistas.travel

Exact-Match SLDs

A different *Domains 4 All* user already has an established brand name for their product. Unfortunately, the domain name for their brand is unavailable for the TLD, “com”. This is a common situation. In response, the user uses the search form tool to perform an exact match SLD search for their brand, “sundogs”, in order to see which TLDs will be suggested.

To support this use case, RNS provides the parameter *exactSldOnly*, as exemplified below.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=sundogs&**exactSldOnly**=1

Search Results:

Rank	Suggestion
1	sundogs.limited
2	sundogs.glass
3	sundogs.dance
4	sundogs.pet
5	sundogs.solar
6	sundogs.forsale
7	sundogs.shop
8	sundogs.protection
9	sundogs.energy
10	sundogs.life

Targeting TLDs by User Location

Parameters Discussed: *clientIP*

Many customers are interested in returning domains that contain TLDs relevant to a user's geographic location. RNS provides robust support for this feature, called *Geotargeted TLDs*. Specifically, given a user's location, RNS will increase the weight of a geotargeted TLD, thereby increasing the overall domain score.

The most common example of Geotargeted TLDs is to return domains containing a country code TLD. For example, if your user is located in Germany, it might be appropriate to return domain recommendations for "de". Furthermore, you might also consider targeting TLDs for specific regions or cities. A user from New York City might be interested in "nyc", while a user from Montreal may be interested in "quebec".

You can easily target geographic TLDs by providing the parameter *clientIP*. All you need to do is supply your user's IP Address, and RNS will do the rest. *Domains 4 All* serves users from around the world, and supporting geotargeted TLDs is very important. To exemplify how RNS solve this need, we can use the same query from The Default Query section, except this time we'll add a user IP Address from France.

Search Query:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input=rnssearch&clientIP=176.31.84.249
```

Search Results:

Rank	Suggestion
1	rnssearch.rocks
2	rnssearch.live
3	rnssearch.site
4	rnssearch.social
5	rnssearch.online
6	rnssearch.xyz

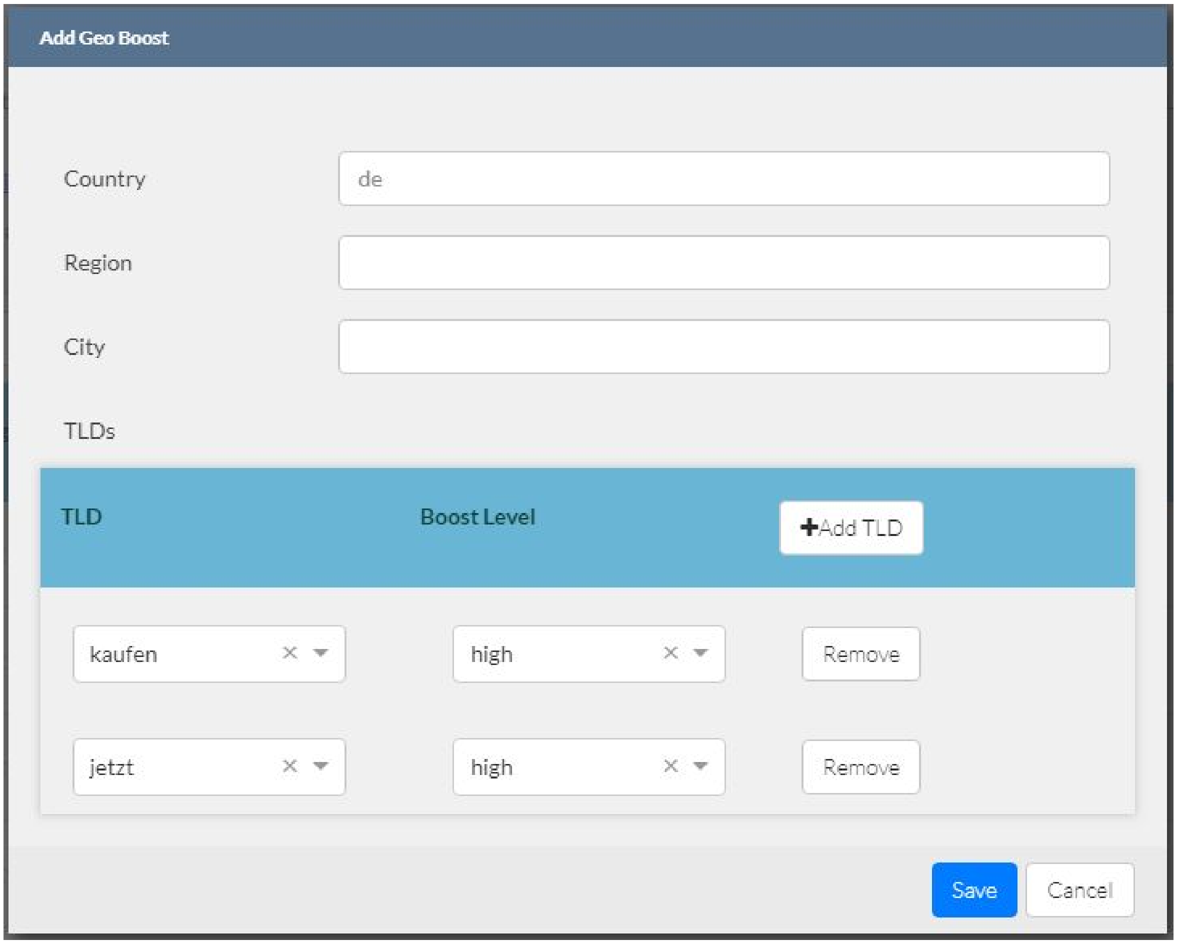
7	**rnssearch.fr**
8	rnssearch.news
9	rnssearch.video
10	rnssearch.pics

RNS Customer Portal

Geotargeted TLDs can be managed using the RNS Customer Portal. Within the portal, you can set:

1. The TLDs that you want targeted.
2. The supported geographic location(s) for a TLD. RNS allows you to target TLDs at the following levels:
    a. Country
    b. Region
    c. City
3. The weight level for a given TLD and geographic location combination. Weight levels can be to high, medium, and low.

In the following example, *Domains 4 All* wants to increase the weight for language-specific TLD, meaning that the search originates from the location of the language. They have chosen to target the TLDs "kaufen” and "jetzt” for all searches coming from Germany. To accomplish this, they would add the following entry on the Geotargeting TLDs page.



Country: de

Region:

City:

TLDs

TLD	Boost Level	
kaufen	high	Remove
jetzt	high	Remove

Please note: It may take up to 20 minutes for the RNS API to reflect changes made in the Customer Portal.

Tracking Query Metrics

Parameters Discussed: *usageId*

Customers call RNS in a variety of ways. For example, they may make several different calls for the same search, tailoring results for different sections of the page. Alternatively, they may have separate domain search flows for targeted users. Differentiating these RNS queries is important for tracking metrics such as impressions, clicks, and conversions. This is especially true when using RNS as a partial solution; customers must know where the data originated from. To solve this problem, RNS provides the *usageId* parameter.

The *usageId* parameter exists solely for you. You provide a value for the parameter, and RNS will return it in the recommended domains output. RNS makes no logic decisions based on the parameter value.

Below, we'll provide two examples of how to use a *usageId*.

Two Queries

In the following example, *Domains 4 All* is making two calls to RNS for the same search. In the first query, they are requesting one relevant domain suggestion for the TLD, "club", which they will display in a specific position as part of a presentment campaign. In the second query, *Domains 4 All* are requesting 20 relevant suggestions, which they'll display in a "Suggestions" section on their results page.

Search Query 1:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input
=eastsideevents&onlyTLDs=club&maxCount=1&usageId=presentmentCampaign-2342
```

Search Results 1 JSON Response:

```
"input": {
   "usageid": "presentmentCampaign-2342"
}
```

Search Query 2:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input
```

```
=eastsideevents&maxCount=20&usageId=suggestionsBox
```

Search Results 2 JSON Response:

```
"input": {
   "usageid": "suggestionsBox"
}
```

A/B Testing

Customers are always trying to improve conversion rates. If you're reading this document, then you're in the same boat. Determining the best results to display and how to display them is a never-ending enterprise. *Domains 4 All* is trying to do just that.

In this second example, *Domains 4 All* uses the RNS *usageId* as a version tracker for their suggestions box. For their A/B tests, they'll send some traffic to Version 13 and the rest of the traffic to Version 14.

Search Query 1:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input
=eastsideevents&maxCount=20&usageId=suggestionsBox-version13
```

Search Results 1 JSON Response:

```
"input": {
   "usageid": "suggestionsBox-version13"
}
```

Search Query 2:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input
=eastsideevents&maxCount=20&usageId=suggestionsBox-version14
```

Search Results 2 JSON Response:

```
"input": {
    "usageid": "suggestionsBox-version14"
 }
```

Premium Domains

Parameters Discussed: *allowPremium, allowPlatinum*

RNS provides a few options for returning premium domains. You may choose to return relevant premium domains to your users, or you may tell RNS to filter them out completely. If you choose the former, then it's important to understand how premium recommendations work in our system.

First, RNS has the ability to return premium domains, but it will only do so if the domains are considered *relevant* to the user input. This means that RNS does not explicitly try to find premium domains. Rather, if RNS organically suggests a domain because its relevant, and if that domain happens to be premium, then the API can return that domain in the result set. In this situation, RNS will always return a flag, marking that the domain in question is premium.

Additionally, RNS does *not* return pricing information with any domain. Since premium domains are typically priced at higher tiers, it is important to be aware of this when displaying results to your users. If you utilize a different user checkout flow for premium domains, then you will need to handle that situation appropriately

Finally, some registries support a domain price tier above premium, which RNS refers to as "platinum". Platinum domains are generally quite expensive, and you may want to handle these differently as well.

In order to support a number of different uses cases for both premium and platinum domains, RNS provides the query parameters *allowPremium* and *allowPlatinum* parameters.

Default Behavior: Exact SLD Match

By default, RNS will only return premium and platinum domains if the SLD of the domain is an *exact match* of the SLD from user search input. In the example below, a *Domains 4 All* user is searching for the value, “sports”. This is a very general input, and there are a few premium and platinum domains recommended in a result.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=sports&**maxCount**=15&**allowPremium**=1&**allowPlatinum**=1

Search Results:

Rank	Suggestion	Premium	Platinum
1	sports.physio	FALSE	FALSE
2	sports.fans	TRUE	FALSE
3	sports.cards	FALSE	TRUE
4	sports.news	FALSE	TRUE
5	sports.bar	FALSE	FALSE
6	sports.games	FALSE	TRUE
7	sports.academy	FALSE	TRUE
8	sports.video	FALSE	TRUE
9	sport.physio	FALSE	FALSE
10	sport.bar	FALSE	FALSE
11	game.physio	FALSE	FALSE
12	game.bar	FALSE	FALSE
13	games.physio	FALSE	FALSE
14	play.bar	FALSE	FALSE
15	games.bar	FALSE	FALSE

Return Any Relevant Domain

RNS supports similar functionality to the default option, where it will return *any* relevant premium or platinum domain for a given user search. Reusing the input, “sports”, from the previous section, notice that significantly more premium and platinum domains are recommended.

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=sports&**maxCount**=15&**allowPremium**=2&**allowPlatinum**=2

Search Result:

Rank	Suggestion	Premium	Platinum
1	sports.physio	FALSE	FALSE
2	sports.fans	TRUE	FALSE
3	sports.cards	FALSE	TRUE
4	sports.news	FALSE	TRUE
5	sports.bar	FALSE	FALSE
6	sports.games	FALSE	TRUE
7	sports.academy	FALSE	TRUE
8	sports.video	FALSE	TRUE
9	sport.physio	FALSE	FALSE
10	sport.fans	TRUE	FALSE
11	sport.news	FALSE	TRUE
12	sport.bar	FALSE	FALSE
13	sport.fitness	TRUE	FALSE
14	sport.games	FALSE	TRUE
15	game.physio	FALSE	FALSE

Return None

RNS provides one final option, which allows you to completely filter out premium or platinum domains from a search. This means that RNS will return zero premium or platinum domains in the result list. Reusing the input, “sports”, from the previous sections, notice how the list of recommended domains changes.

Search Query:

```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input
=sports&maxCount=15&allowPremium=0&allowPlatinum=0
```

Search Result:

Rank	Suggestion	Premium	Platinum
1	sports.physio	FALSE	FALSE
2	sports.bar	FALSE	FALSE
3	sport.physio	FALSE	FALSE
4	sport.bar	FALSE	FALSE
5	game.physio	FALSE	FALSE
6	game.bar	FALSE	FALSE
7	games.physio	FALSE	FALSE
8	play.bar	FALSE	FALSE
9	games.bar	FALSE	FALSE
10	allsports.cards	FALSE	FALSE
11	allsports.bar	FALSE	FALSE
12	sportsclub.physio	FALSE	FALSE
13	sportsclub.fans	FALSE	FALSE
14	allsports.fitness	FALSE	FALSE
15	sportsclub.cards	FALSE	FALSE

Adult and Offensive Names

Parameters Discussed: *allowAdult, allowOffensive*

Customers have different target markets, and they may need to be careful about the types of domains that are displayed. This is especially true if you want to attract a younger or more family-friendly audience. To support this type of need, RNS provides the ability to filter out adult and offensive words from your result list.

In the following query, *Domains 4 All* shows how to use the *allowAdult* and *allowOffensive* filters, so that domains containing these words will not be returned.

Search Query:

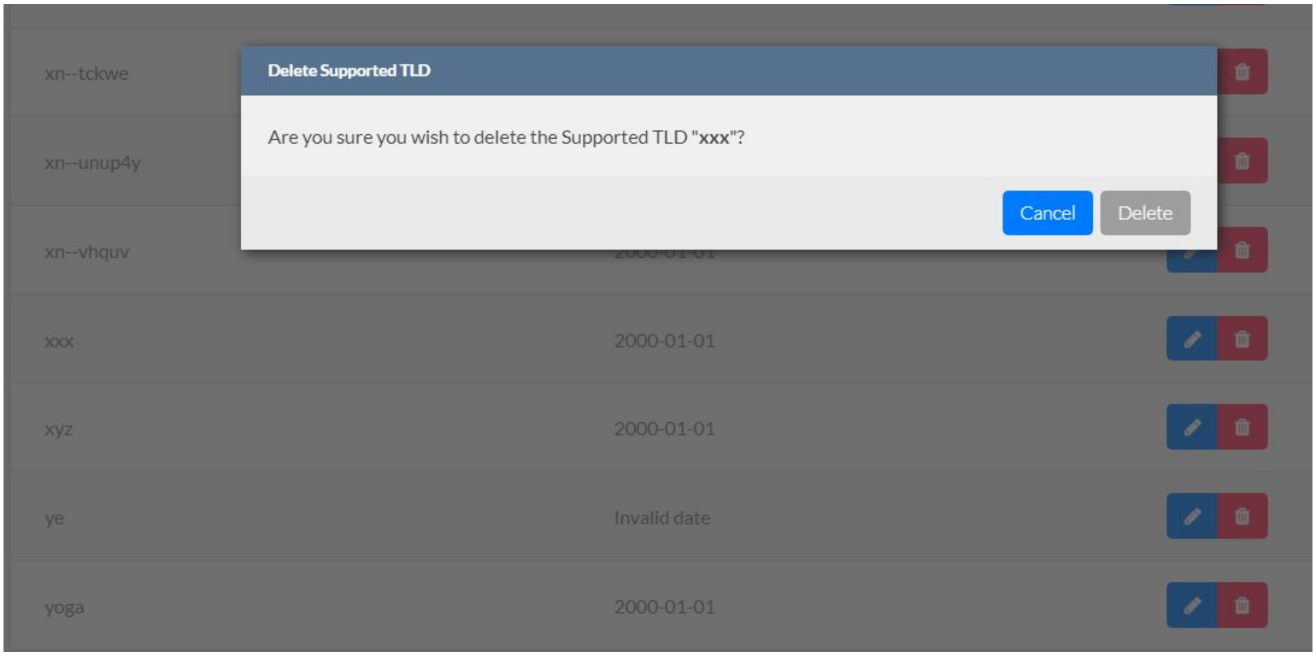
```
https://api.rns.domains/recommend-domains?account=domains4all&password=abcd123&input=rnssearch&allowAdult=0&allowOffensive=0
```

Supporting Your TLDs

RNS is capable of recommending domains for almost 1000 TLDs, empowering our service to be as relevant as possible. As a result, we encourage customers to support as many TLDs as possible for the benefit of their users. However, we do understand that each customer has different capabilities and use cases. In many scenarios, customers may only be licensed for a certain set of TLDs.

To address this need, RNS allows you to configure the specific TLDs that you want to support. This means that, for all queries, RNS will only return domains for your supported TLDs. Please note that when a new account is created for you, TLDs will automatically be associated with your account, so that you can start using our service immediately. In order to configure your supported TLDs, access our Customer Portal, and navigate to RNS Features > Supported TLDs.

In an example, *Domains 4 All* decided that it did not want to return adult and offensive domain suggestions. As a result, for all of their queries, they pass a value of *0* for *allowAdult* and *allowOffensive*. Taking this one step further, they have also decided not to display RNS suggestions for the TLD, “xxx”. To do this, *Domains 4 All* logs into the RNS Customer Portal and navigates to the Supported TLDs page. Finding the TLD, “xxx”, they remove it from the list. Going forward, queries will no longer recommend domains for this TLD.



Delete Supported TLD

Are you sure you wish to delete the Supported TLD **"xxx"**?

xn--tckwe	
xn--unup4y	
xn--vhquv	2000-01-01
xxx	2000-01-01
xyz	2000-01-01
ye	Invalid date
yoga	2000-01-01

Hints: Custom Keyword Categories

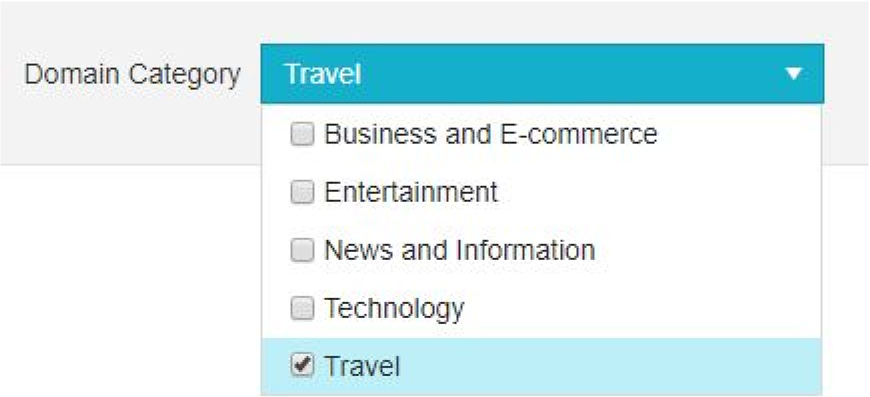
Parameters Discussed: *hints*

RNS provides an advanced feature for influencing TLD weights called *Hints*. Hints allow you to define a keyword “context” for a given user search. Within this context, the weights of TLDs in a category will be increased, raising the final score and rank of the associated recommended domains. Hints is an incredibly capable yet complicated feature which requires some upfront configuration. In the sections below, we’ll go through an example of how you might utilize this functionality.

User Interface

Let’s say that a *Domains 4 All* user is searching for a domain name for their new blog. Furthermore, the user knows that they want to write about travel-related activities. In this example, the content of the domain falls under a general category of “travel”.

Additionally, *Domains 4 All* provides a number of UI search features for their users. One of these features is called “Domain Category”. When a category is selected, then the weights of specified TLDs will be *increased* in the search results. In our example, the user would most likely choose the category, “Travel”.



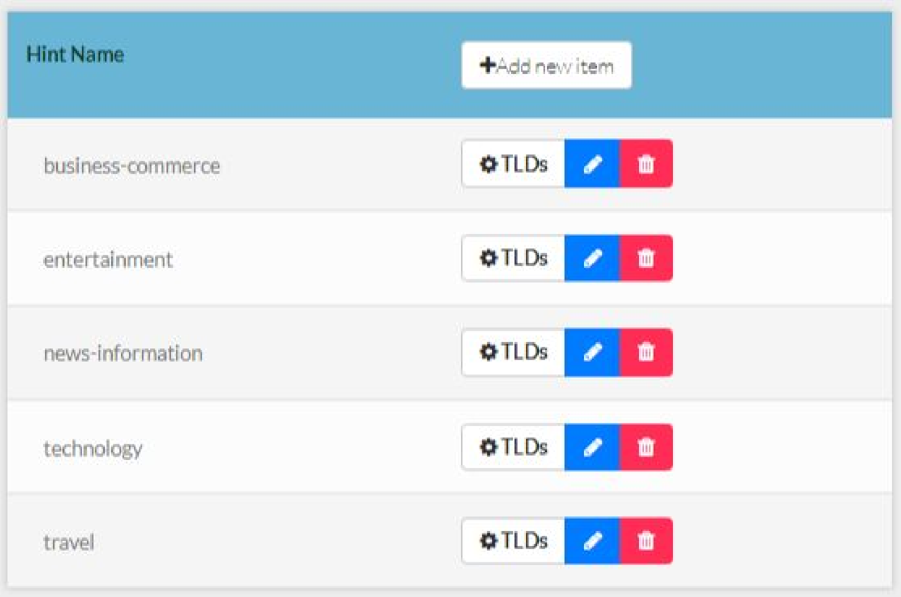
Domain Category: Travel ▼

- □ Business and E-commerce
- □ Entertainment
- □ News and Information
- □ Technology
- ☑ Travel

Customer Portal

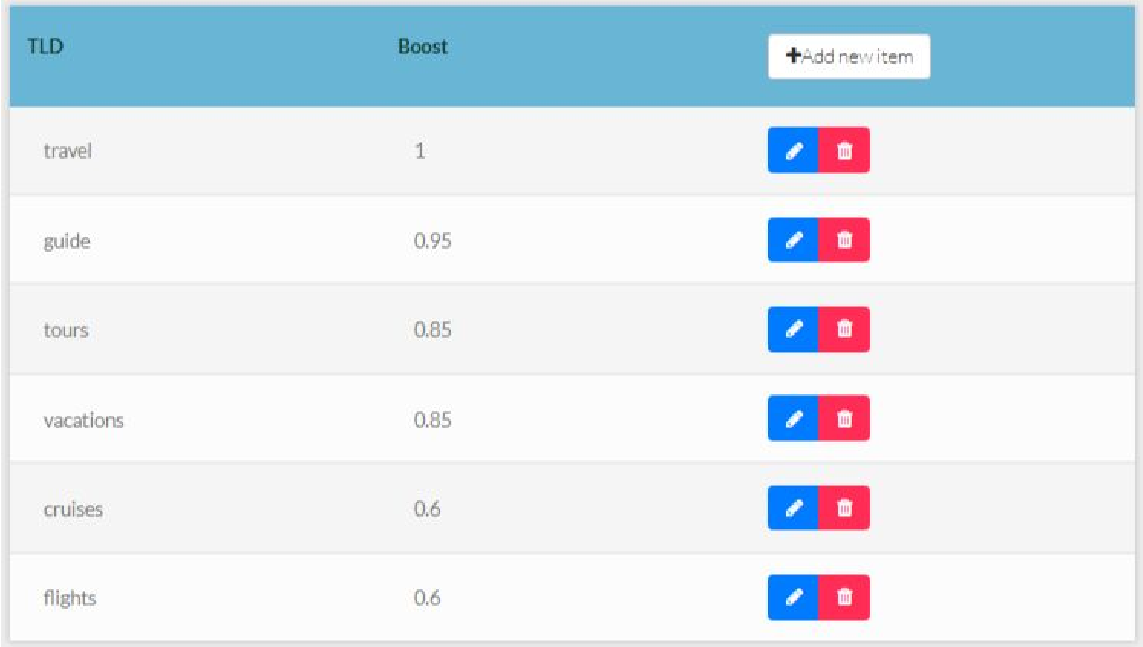
Continuing our example, in order to support the concept of a “Domain Category”, *Domains 4 All* must first set the TLDs and weights for each domain category that they want to support. This configuration is accomplished via the *Hints* page in the RNS Customer Portal. Navigate to RNS Features > Hints.

In the image below, *Domains 4 All* has created a *hint* for each category on their UI.



Hint Name	+Add new item
business-commerce	⚙TLDs
entertainment	⚙TLDs
news-information	⚙TLDs
technology	⚙TLDs
travel	⚙TLDs

Next, *Domains 4 All* selects the TLDs and weight levels they want associated with each hint. Below, *Domains 4 All* has configured the following TLDs for the “travel” category:



TLD	Boost
travel	1
guide	0.95
tours	0.85
vacations	0.85
cruises	0.6
flights	0.6

Please note: It may take up to 20 minutes for the RNS API to reflect changes made in the Customer Portal.

API Query

Finally, *Domains 4 All* must tell RNS to use a specific hint when an associated category is selected via the UI. To accomplish this, *Domains 4 All* must provide the *hints* parameter in the API query, and they must supply the hint value configured in the RNS Customer Portal.

In our example, the user has selected the Domain Category, “travel”, and they have entered the search term, “ElevatedLife”. The query for this might look as follows:

Search Query:

https://api.rns.domains/recommend-domains?**account**=domains4all&**password**=abcd123&**input**=elevatedlife&**hints**=travel

Search Results:

In order to illustrate the power of the Hints feature, let’s look at side-by-side results comparing suggestions *without* and *with* hints. As you can see below, hints significantly impacts the domains recommended, returning more TLDs related to the category, “travel”.

Without Hints		**With** Hints	
Rank	Suggestion	Rank	Suggestion
1	elevatedlife.style	1	elevatedlife.travel
2	elevatedlife.coach	2	elevatedlife.style
3	elevated.social	3	elevatedlife.guide
4	elevated-life.style	4	elevatedlife.vacations
5	elevatedlife.social	5	elevatedlife.tours
6	elevatedlife.fitness	6	elevatedlife.coach
7	elevatedlife.gives	7	elevatedlife.cruises
8	elevated-life.coach	8	elevatedlife.flights
9	elevatedlife.health	9	elevated-life.travel
10	elevatedlife.church	10	elevated.social