StrongLoop API

The StrongLoop API lets you create your own plugins, scripts, and automated methods to interact and integrate with StrongLoop's note-taking database. You can use it to integrate your stored notes into another project or website, or update your note database through a shell or script.

Strongloop uses the REST protocol for its API functions, so if you are familiar with other APIs like Twitter API or YouTube API, you should be comfortable with its methods.

Before You Begin

StrongLoop has no authentication requirements, so no configuration is required. You can invoke REST commands directly through your browser using paths and query strings.

StrongLoop provides the following response types for all its methods:

- application/json
- application/xml
- application/javascript
- text/xml
- text/javascript

Methods

/Notes

» GET

Searches and returns all note entries that match the supplied filter. The filter uses SQL conditionals `where`, `include`, `order`, `offset`, and `limit`. The filter must be enclosed with curly braces. The response returns an array of entries enclosed in square braces.

Request `http://138.197.146.79/api/Notes?filter={filter}`

Parameter	Description	Type
filter	SQL filter	Code

Example:

```
{"where":{"title":"Dylan's phone number"}}
```

Response

Code	Description
200	GET request succeeded.
400	Syntax error. GET request failed.

Example:

```
[
    {
        "title": "Dylan's phone number",
        "content": "555-1111",
        "id": 21
    }
]
```

» PUT

Replaces a note entry, or if it doesn't exist, create a new one with the supplied parameters.

Request `http://138.197.146.79/api/Notes`

Parameter	Description	Type
title	Name of entry	String
content	Value of entry	String

Example:

```
{
    "title": "Dylan's phone number",
    "content": "555-2222"
}
```

Response

Code	Description
200	PUT request succeeded.
400	Syntax error. GET request failed.
422	Missing or bad parameters. PUT request failed.

Example:

```
{
    "title": "Dylan's phone number",
    "content": "555-2222",
    "id": 21
}
```

» POST

Create a new note entry.

Request `http://138.197.146.79/api/Notes`

Parameter	Description	Type
title	Name of entry	String
content	Value of entry	String

Example:

```
{
    "title": "Alex's phone number",
    "content": "555-3333"
}
```

Response

Code	Description
200	POST request succeeded.
400	Syntax error. POST request failed.

Code	Description
422	Missing or bad parameters. POST request failed.

Example:

```
{
    "title": "Alex's phone number",
    "content": "555-3333",
    "id": 22
}
```

/Notes/{id}

» GET

Search for an entry by its unique identifier.

Request `http://138.197.146.79/api/Notes/{id}`

No parameters.

Response

Code	Description
200	GET request succeeded.
404	ID not found.

Example:

```
{
    "title": "Dylan's phone number",
    "content": "555-1111",
    "id": 21
}
```