**Forecast API Specification Doc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 15-Oct-2021 | Davide Losco | Initial draft |

## 

## 

## 

Index

1. Methods

* Login
* Get city forecast
* Set city forecast
* Delete city forecast

1. Glossary

* Conventions
* Status Codes
* Object Structure

## 

## 

## 

Methods

## 1. login

Authenticate the user with the system and obtain the auth\_token

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/v{v}/login/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Descr** |
| string | api\_key | The api\_key helps the server to validate the request source. |
| string | username | username |
| string | password | password |
| string | version | The current version of internal recipe database. |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | {  "auth\_key": <auth\_key>  }  auth\_key (**string**) - all further API calls (except GET) must have this  key in header |
| 403 | {"error":"API key is missing."} |
| 400 | {"error":"Please provide username."} |
| 400 | {"error":"Please provide password."} |
| 401 | {"error":"Invalid API key."} |
| 401 | {"error":"Incorrect username or password."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 2. get city forecast

Get the forecast of a city for a date

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/v{v}/forecast/ |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Descr** |
| string | uuidCity | The universally unique identifier of the city for which the forecast is required |
| Date | dateF | The date for which forecast is required |
| string | version | The current version of internal recipe database |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be an object forecast containing the detail of**  **requested Forecast.**  Example response:  {  "date": "2021-10-15",  "date\_epoch": 1634256000,  "day": {  "maxtemp\_c": 15.3,  "maxtemp\_f": 59.5,  "mintemp\_c": 10.5,  "mintemp\_f": 50.9,  "avgtemp\_c": 12.7,  "avgtemp\_f": 54.9,  "maxwind\_mph": 9.2,  "maxwind\_kph": 14.8,  "totalprecip\_mm": 0.0,  "totalprecip\_in": 0.0,  "avgvis\_km": 10.0,  "avgvis\_miles": 6.0,  "avghumidity": 68.0,  "daily\_will\_it\_rain": 0,  "daily\_chance\_of\_rain": 0,  "daily\_will\_it\_snow": 0,  "daily\_chance\_of\_snow": 0,  "condition": {  "text": "Partly cloudy",  "icon": "//cdn.weatherapi.com/weather/116.png",  "code": 1003  },  "uv": 2.0  }  } |
| 400 | {"error":"Please specify database version."} |
| 400 | {"error":"Invalid database version."} |
| 400 | {"error":"City requested not found."} |
| 400 | {"error":"Valid date is requested."} |
| 401 | {"error":"Invalid API key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 3. set forecast city

Insert or update the forecast of a day for a city

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/v{v}/forecast |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Descr** |
| string | auth\_key | Login authentication key |
| forecast | forecast | Forecast obj |
| Date | dateF | The date for which forecast will be set |
| string | uuidCity | The universally unique identifier of the city for which the forecast will be set |
| string | version | The current version of internal  recipe database |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be a boolean that indicate the result of the operation**  **and a string that specified if it was an insert or an update**  Example response:  {  "result":true,  "operation":"Insert"  } |
| 400 | {"error":"Please specify database version."} |
| 400 | {"error":"Invalid database version."} |
| 400 | {"error":"Invalid date for forecast."} |
| 400 | {"error":"City not found."} |
| 400 | {"error":"Invalid forecast."} |
| 401 | {"error":"Invalid Auth key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

## 4. delete forecast

Delete the forecast of a day for a city

## Request

|  |  |
| --- | --- |
| **Method** | **URL** |
| **DELETE** | api/v{v}/forecast |

|  |  |  |
| --- | --- | --- |
| **Type** | **Params** | **Descr** |
| string | auth\_key | Login authentication key |
| Date | dateF | The date for which forecast will be set |
| string | uuidCity | The universally unique identifier of the city for which the forecast will be set |
| string | version | The current version of internal  recipe database |

## Response

|  |  |
| --- | --- |
| **Status** | **Response** |
| 200 | **Response will be a boolean that indicate the result of the operation**  Example response:  {  "result":true,  } |
| 400 | {"error":"Please specify database version."} |
| 400 | {"error":"Invalid database version."} |
| 400 | {"error":"Invalid date for forecast."} |
| 400 | {"error":"City not found."} |
| 401 | {"error":"Invalid Auth key."} |
| 500 | {"error":"Something went wrong. Please try again later."} |

Glossary

## Conventions

* All response are in JSON format.
* All request parameters are mandatory

## Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |

## Objects Structure

Objects used as parameter for set request and as party of the response in get request

## Forecast

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| date | string | Forecast date |
| date\_epoch | int | Forecast date as unix time. |
| day | element | See day element |

## Day

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| maxtemp\_c | decimal | Maximum temperature in celsius for the day. |
| maxtemp\_f | decimal | Maximum temperature in fahrenheit for the day |
| mintemp\_c | decimal | Minimum temperature in celsius for the day |
| mintemp\_f | decimal | Minimum temperature in fahrenheit for the day |
| avgtemp\_c | decimal | Average temperature in celsius for the day |
| avgtemp\_f | decimal | Average temperature in fahrenheit for the day |
| maxwind\_mph | decimal | Maximum wind speed in miles per hour |
| maxwind\_kph | decimal | Maximum wind speed in kilometer per hour |
| totalprecip\_mm | decimal | Total precipitation in milimeter |
| totalprecip\_in | decimal | Total precipitation in inches |
| avgvis\_km | decimal | Average visibility in kilometer |
| avgvis\_miles | decimal | Average visibility in miles |
| avghumidity | int | Average humidity as percentage |
| [condition:text](https://www.weatherapi.com/docs/#weather-icons) | string | Weather condition text |
| [condition:icon](https://www.weatherapi.com/docs/#weather-icons) | string | Weather condition icon |
| [condition:code](https://www.weatherapi.com/docs/#weather-icons) | int | Weather condition code |
| uv | decimal | UV Index |