Key-Based Authentication

This lesson gives an overview of a class of web APIs that require key based authentication.

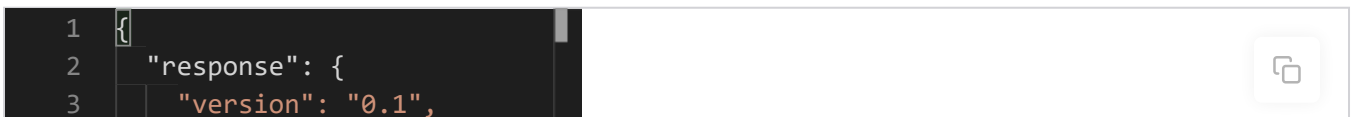
Another class of APIs requires the client to authenticate himself when accessing the service. Authentication can be done via several techniques. In this paragraph, we'll use the simplest one: access key. An *access key* is a generated string containing characters and digits and associated to a user.

> Of course, authentication-based APIs often also have rate limits.

There is no universal standard regarding access keys. Each service is free to use its own custom format. The client must provide its access key when accessing the API, generally by adding it at the end of the API URL. A prerequisite for using any key-based web API is to generate oneself an access key for this particular service.

Let's put this into practice for obtaining about the current weather in your area. To do so, you could simply look outside the window, but it's way cooler to use the Weather Underground web service instead. This service has a key-based API for retrieving the weather in any place. To obtain it, you'll have to sign up as a user (it's free) and generate a new API key by registering your application.

Once you've done this, weather data is available through an URL of the form http://api.wunderground.com/api/ACCESS_KEY/conditions/q/COUNTRY/TOWN.json. Replace `ACCESS_KEY`, `COUNTRY` and `TOWN` with your own settings, and you should obtain the weather in your surroundings. The necessary first step is to check out and understand the API data format. The result of an API call looks like this when getting weather for Bordeaux, France.



```
    "termsofService": "http://
    "features": {
      "conditions": 1
    }
  },
  "current_observation": {
    "image": {
      "url": "http://icons.wxug
      "title": "Weather Undergr
      "link": "http://www.wunde
    },
    "display_location": {
      "full": "Bordeaux, France
      "city": "Bordeaux",
      "state": "33",
      ...
    },
    "observation_location": {
      "full": "Bordeaux, ",
      "city": "Bordeaux",
      "state": "",
      "country": "FR",
      ...
    },
    "estimated": {},
    "station_id": "LFBD",
    "observation_time": "Last U
```


Now we just have to call the API from our JavaScript code and display the main result on a web page.

Output

JavaScript

HTML

```
/*
Weather in your area
*/

// Please generate yourself an
fetch(
  "http://api.wunderground.com/
)
  .then(response => response.js
  .then(weather => {
    // Access some weather prop
    const location = weather.cu
    const temperature = weather
    const humidity = weather.cu
    const imageUrl = weather.cu
    // Create DOM elements for
    const summaryElement = docu
    summaryElement.textContent
```

```
19      const imageElement = docume
20      imageElement.src = imageUrl
21      // Add location to title
22      document.querySelector("h2"
23      // Add elements to the page
24      const weatherElement = docu
25      weatherElement.appendChild(
26      weatherElement.appendChild(
27    })
28    .catch(err => {
29      console.error(err.message);
30    });
```