

Notre solution - Classement

Par Ahmad Charba et Simon Proulx

Les technologies utilisées

- PHP
- JSON
- API
- PostgreSQL
- Apache
- Windows
- Chromium

JSON

- Data-format that is user-friendly and easy to parse for the computer
- Subset of ECMAScript
- similar to XML
- real-time server-to-browser protocol

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

JSON encode/decode

- we use JSON encode/decode as a “hack” to transform a dictionary/array as an object.

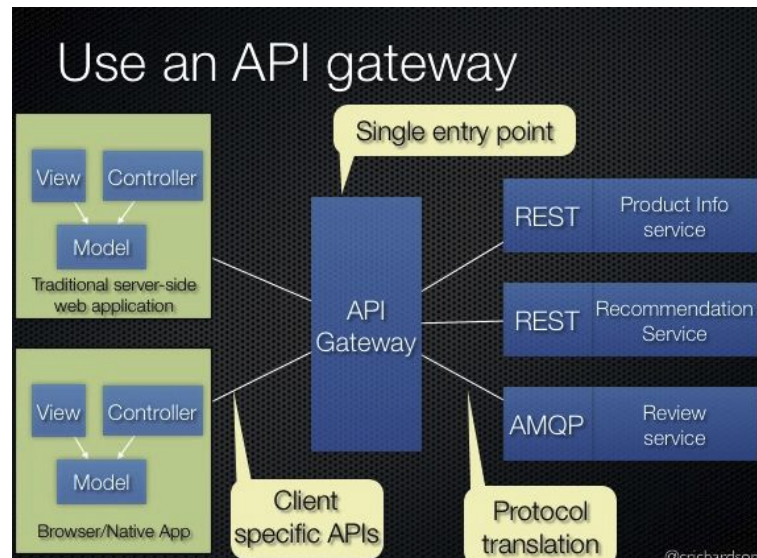
```
<?php
require 'api/contact_api.php';

$athletes = JSON_decode(get_contact($_GET['id']));

echo '<table class="my_table">
    <tr>
        <th>Gender</th>
        <th>Name</th>
        <th>Email</th>
        <th>Phone Number</th>
```

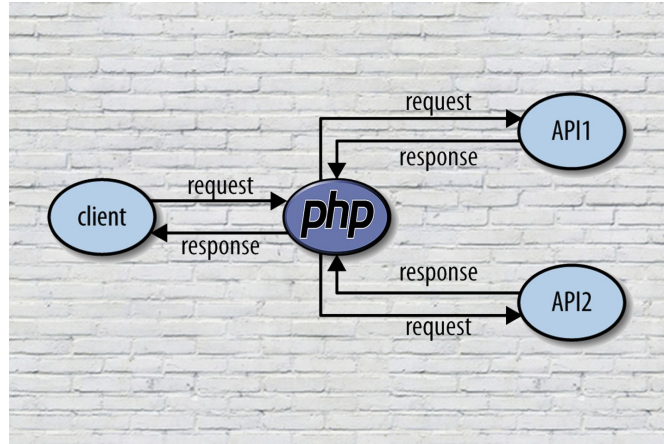
APIs en PHP

- Client to query connection
- ended up not using a micro-service as authentication (tokens) ended up not being a concern



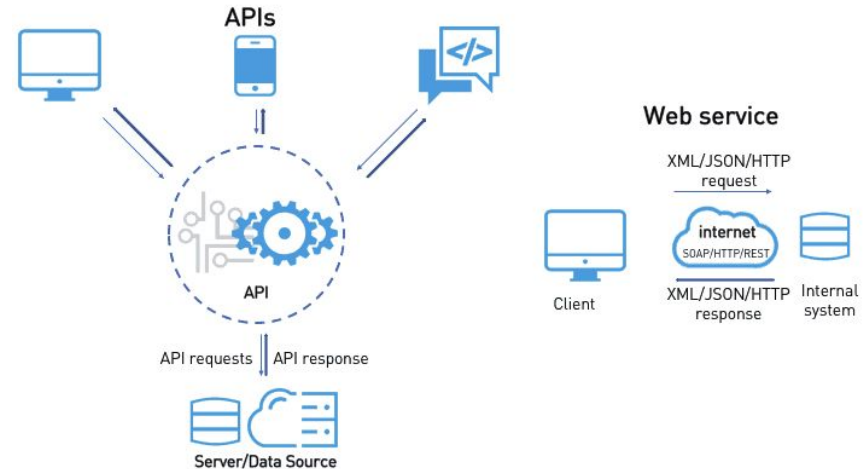
Using API Client

- Guzzle is a service we have considered using
- Guzzle is an HTTP client for PHP, allows for connection with API very easily
- API client allows for better security within the website code
- Clients can allow modulation over the API's interaction, meaning multiple clients may be allowed on the server without any issues



Advantages du API mentionner

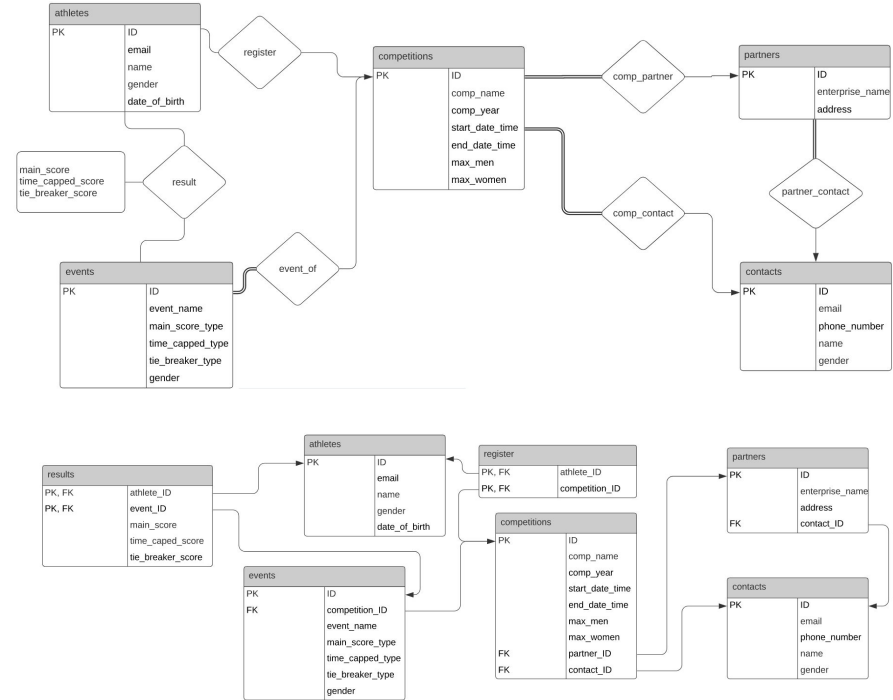
- API makes it safe to access the queries
- PHP, HTML and SQL queries are separated for better visual and structured code
- unified API allows for shorter code
- easy to include micro-services



Back end Architecture

- Database created using PostgreSQL
- Usage of Constraints such as UNIQUE, NOT NULL and PRIMARY KEY to maintain table's integrity
- Usage of VIEWS, which the API takes the data from without having access to the rest of the information

Diagrams:



Website architecture

- Using page-by-page format for simplicity
- Every table page is associated with an SQL query
- Leaderboard has complex php code and SQL query
- Compétitions, Partners -> Contact
- Athlete, Competitions -> Events

