# Advertisement Request Enhancer

We would like you to construct a data-processing pipeline that augments an incoming Ad request with relevant contextual information. For the most part, this will involve fetching data from various microservices and injecting it into the Ad request.

We recommend focusing on the Primary Objectives first, and then gradually improving your solution to incorporate the bonus objectives.  From our standpoint, we primarily want to start a conversation around how you approach the problem, and then discuss how you would like to approach the unimplemented objectives.

Feel free to implement your solution in either: node.js / Ruby / Java / Python.

### Primary Objectives

- Create a microservice endpoint that enhances an incoming Advertisement Request with additional contextual information.
    - Inject the Site Demographics.
    - Inject the Publisher details.
    - Inject the Country of Origin.
- Demonstrate the latency added < 500ms.
- Provide automated tests to demonstrate the correctness of your implementation.

### Bonus Objectives (attempt at least 2)

- How fast can you make the end to end execution? Demonstrate the latency of your application.
- Ensure that the application can handle an average of 50 requests per second over an extended period of time.
- If request originates from an IP address outside of the United States, then abort the transaction before calling any internal web services and respond with an error message.
- If the Publisher ID cannot be obtained, then abort the transaction.
- Allow individual processing units to easily be installed / uninstalled at runtime.
- Provide ways to monitor the health and performance of the application at runtime.
- Ensure that the application is fault-tolerant.  That is, the end-to-end processing of an incoming request should be able to proceed despite the failure of a non-required service.

# Data Schema

### Incoming Requests

An incoming Advertisement Request will only contain the following fields:

| Field | Required? | Purpose |
|---|---|---|
| site.id | Yes | A unique identifier for a specific web site. |
| site.page | Yes | The full URL for the page that yielded the Advertisement Request. |
| device.ip | Yes | The IP address of the device that visited the page. |
| user.id | No | An identifier that tracks the user across page visits. |

For example:

```
{
  "site": {
    "id": "foo123",
    "page": "http://www.foo.com/why-foo"
  },
  "device": {
    "ip": "69.250.196.118"
  },
  "user": {
    "id": "9cb89r"
  }
}
```

**Outgoing Responses**

After an Advertisement Request has been augmented, it may contain the following additional fields:

| Field | Required? | Purpose |
|---|---|---|
| site.demographics.female_percent | No | The percentage of users during a 24 hour period that are female. |
| site.demographics.male_percent | No | The percentage of users during a 24 hour period that are male. |
| publisher.id | Yes | The ID of the Publisher that owns the Site. |

| | | |
|---|---|---|
| publisher.name | No | The name of the Publisher that owns the Site. |
| device.geo.country | No | The country in which the device's IP address resides. |

For example:

```
{
  "site": {
    "id": "foo123",
    "page": "http://www.foo.com/why-foo",
    "demographics": {
      "female_percent": 62,
      "male_percent": 38
    },
    "publisher": {
      "id": "2359sdf",
      "name": "Foo LLC"
    }
  },
  "device": {
    "ip": "69.250.196.118",
    "geo": {
      "country": "US"
    }
  },
  "user": {
    "id": "9cb89r"
  }
}
```

# Resources

| | |
|---|---|
| Publisher Lookup Service | **Documentation:**<br>http://159.89.185.155:3000/apidoc/ |

| | |
|---|---|
| | **API:**<br>POST http://159.89.185.155:3000/api/publishers/find<br>The Site -> Publisher Mapping rarely changes. In fact, it basically only changes when one Publisher buys out another Publisher. |
| Demographics Lookup Service | **Documentation:**<br>http://159.89.185.155:3000/apidoc/<br>**API:**<br>GET http://159.89.185.155:3000/api/sites/SITE_ID/demographics<br>The per-Site demographics data is typically refreshed on a weekly basis. |
| Geo-IP Lookup | We recommend using a library that is based upon the MaxMind data (free trial account should work):<br>https://dev.maxmind.com/geoip/<br>Keep in mind: there are many 3rd-party libraries that are built around this data. So, remember to check your favorite package repository! |

P.S. Some of these microservice are ultimately owned by 3rd-parties, and therefore its behavior is not fully under our control. So, you should expect occasional delays in the response, disruptions, etc.