LIBERTY BIKES

An online game for 1-4 players

Built on microservices
What is Liberty Bikes?
What is Liberty Bikes?

- Objective: be the last player alive
- Bikes move at the same speed
- Bikes leave a trail
- Players steer up/down/left/right
- Hit any obstacle, trail, or player, and you’re out!

- Join a game by entering a code
- Each round lasts about 1 minute
- Stats for each player are tracked across rounds
- Play a few games, then come back later to see if you still hold the top spot
Overall Design

- UI (Angular)
- DB
- Player Service (REST)
- Game Service (REST + websockets)
- Auth Service (REST)
- SSO providers (Google, Twitter, GitHub)

Non-Java frontend
Overall Design

- Player Service (REST)
- Auth Service (REST)
- Game Service (REST + websockets)
- UI (Angular)
- DB

CPU + I/O intensive

SSO providers (Google, Twitter, GitHub)
Overall Design

DB

UI (Angular)

Player Service (REST)

Game Service (REST + websockets)

Auth Service (REST)

SSO providers (Google, Twitter, GitHub)

Internal REST calls
Overall Design

Persistent data

DB

Player Service (REST)

Auth Service (REST)

Game Service (REST + websockets)

UI (Angular)

SSO providers (Google, Twitter, GitHub)
Overall Design

UI (Angular)

Player Service (REST)

Game Service (REST + websockets)

Auth Service (REST)

SSO providers (Google, Twitter, GitHub)

External REST calls

DB
Overall Design

UI (Angular)

- Player Service (REST)
- Game Service (REST + websockets)
- DB
- Auth Service (REST)

SSO providers (Google, Twitter, GitHub)

Single-Sign On auth
Benefits of Microservices Architecture

- Independently scale services
Benefits of Microservices Architecture

- Independently scale services
- Develop services independently

![Diagram of microservices architecture](image)

- **UI (Angular)**
- **DB**
- **Player Service v1 (REST)**
- **Game Service (REST + websockets)**
- **Auth Service (REST)**
- **SSO providers**
Benefits of Microservices Architecture

- Independently scale services
- Develop services independently
- Continuous delivery/deployment

- DB
- UI (Angular)
- Auth Service (REST)
- Game Service (REST + websockets)
- SSO providers
- Player Service (REST)
Benefits of Microservices Architecture

- Independently scale services
- Develop services independently
- Continuous delivery/deployment
- Language/framework freedom
Benefits of Microservices Architecture

- Independently scale services
- Develop services independently
- Continuous delivery/deployment
- Language/framework freedom
- Fault/security isolation
Difficulties of Microservices Architecture

- Lots of service-service calls
- More points of failure
- Reconfigure for dev/prod
- Handling authentication

- MP Rest Client + JAX-RS
- MP Fault Tolerance
- MP Config
- MP JWT
The tech stack

- MicroProfile
- Jakarta EE
- Open Liberty
- OpenJDK
- OpenJ9

gradle
Open Liberty

A lightweight open source server runtime ideal for building Java™ microservices and cloud-native apps

- Easy to consume
- Deploy on any cloud for Java™
- Seamlessly transition to WebSphere

https://openliberty.io/
Configure required features

Liberty server.xml includes the needed features
Rest Client

GET https://github.com/login/oauth/access_token?client_id=foo&client_secret=foo&...
MicroProfile Rest Client

GET https://github.com/login/oauth/access_token?client_id=foo&client_secret=foo&…

-Dorg.libertybikes.auth.service.github.GitHubOAuthAPI/mp-rest$url=https://github.com
MicroProfile Fault Tolerance

- @Retry
- @Timeout
- @CircuitBreaker
- @Fallback
MicroProfile Config

```
@Inject
@ConfigProperty(name = "auth_url", defaultValue = "https://localhost:8482/auth-service")
public String authServiceURL;

@Inject
@ConfigProperty(name = "github_key")
public String github_key;
```

Optional

Required

$ export auth_url=https://production.com/auth-service

$ export github_key=superSecret

Override

Validation

java.util.NoSuchElementException: CWMCG0015E: The property foo was not found in the configuration
JSON Web Tokens (JWT)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ

VfiYwvArp2INV6UgpwqqrqfbJp9QpMdv07M8ZI4u4VkwliFtz67lkDF8keyGciql

header

payload

signature

{  "alg": "RS256",  "typ": "JWT"
}

{  "iss": "libertybikes.com",  "sub": "Andy",  "groups": [ "admin", "user" ],  ...
}

RS256(  BASE64(header),  BASE64(payload),  secret)
Many different libs
Each supports different claims
MicroProfile JWT

**BASIC**

```java
@Path("/rank")
@ApplicationScoped
public class RankingService {
  
  @POST
  @RolesAllowed("admin")
  @Path("/{playerId}")
  public void recordGame(@PathParam("playerId") String id, @QueryParam("place") int place) {
    // Can't get here unless 'admin' role
  }
}
```

**CUSTOM**

```java
@Inject
JsonWebToken jwt;

@GET
@Path("/jwt")
public HashMap<String, String> getUserById() {
  String id = jwt.getClaim("id");
  if (db.exists(id)) {
    map.put("exists", true);
    map.put("username", db.get(id).name);
  } else {
    map.put("exists", false);
  }
  return map;
}
```
MicroProfile OpenAPI

```java
@Path("/player")
@ApplicationScoped
@Produces(MediaType.APPLICATION_JSON)
public class PlayerService {

    @GET
    public Collection<Player> getPlayers() {
        // ...
    }

    @Produces(MediaType.TEXT_HTML)
    public String createPlayer(@QueryParam("name") String name,
                               @QueryParam("id") String id) {
        // ...
    }

    @GET
    @Path("/{playerId}")
    public Player getPlayerById(@PathParam("playerId") String id) {
        // ...
    }

    <featureManager>
        <!-- ... -->
        <feature>mpOpenAPI-1.0</feature>
    </featureManager>

    <Server>
        <local:localhost:8081/openapi/ui/>
        http://localhost:8081
    </Server>
    <default>
        GET /player
        POST /player
        GET /player/{playerId}
    </default>
```
Where To Get It

You can play hosted Liberty Bikes here:

(DE) libertybikes-frontend.eu-de.mybluemix.net
(UK) libertybikes-frontend.eu-uk.mybluemix.net
(US) libertybikes.mybluemix.net

Or, clone our repository and run it locally:

github.com/liberty-bikes/liberty-bikes