

Family Map Server: Design and Implementation Tips

CS 240 – Advanced Programming Concepts

Recommended FMS Implementation Steps

1. Model Classes (not really anything to test, especially if your IDE generates most of the code)
2. DAO classes and JUnit tests
3. Main server with File Handler for serving the Web API test page
 - Needed to test the rest
4. JSON Serializer (Object Encoder/Decoder) and Web API (Handler, Service, Request and Response classes) for /clear, /load, and /user/login with JUnit Tests
5. Ancestor generation logic with JUnit Tests
6. The Rest of the Web API with JUnit Tests

Json Deserialization

- A generic Gson deserialization method that will return a runtime specified data type

```
public static <T> T deserialize(String value, Class<T> returnType) {  
    return (new Gson()).fromJson(value, returnType);  
}
```

- **Called by specifying the return type:**

```
MyDataType value = JsonSerializer.deserialize(jsonString, MyDataType.class);
```

Generating a Random Character String with Full Control

- Put available characters in an array
 - Upper and lower case letters
 - Digits
 - Special characters (be careful not to include any Json syntax characters)
- Randomly generate numbers between 0 and array.length

```
import java.util.Random;
...
Random random = new Random();
...
random.nextInt(array.length)
```
- Use the randomly generated numbers to pick characters from the array to be included in the String

Generating a Random Character String with UUID

```
import java.util.UUID;

public class UUIDGenerator {
    public static void main(String [] args) {
        String uuid =
            UUID.randomUUID().toString();
        System.out.println(uuid);
    }
}
```

Request Handlers

- Use inheritance to avoid duplicate code
- Handlers to consider:
 - RequestHandler
 - Top-Level parent handler
 - Code to write a Json response
 - Other code???
 - AuthorizingRequestHandler
 - Parent for handlers that require authorization
 - Read the authorization header to get the auth_token. Return a 401 if missing or invalid
 - PostRequestHandler
 - Parent for handlers that handle post requests
 - Read the request Json and convert to a Java object
 - Other code???
 - Others???

Request Handler Implementation: File Handler

- See `HttpHandler Examples` and `Writing a File Handler` slides in `WebAPI` slides
- For the `Family Map Server` project, in addition to returning a 404 for “File Not Found”, also return the “`/HTML/404.html`” document provided in the files for the server project

Response Class

- Parent class for common response attributes
- 'message' instance variable and getter/setter
- 'success' instance variable and getter/setter