

Class Introduction

1. Introduction

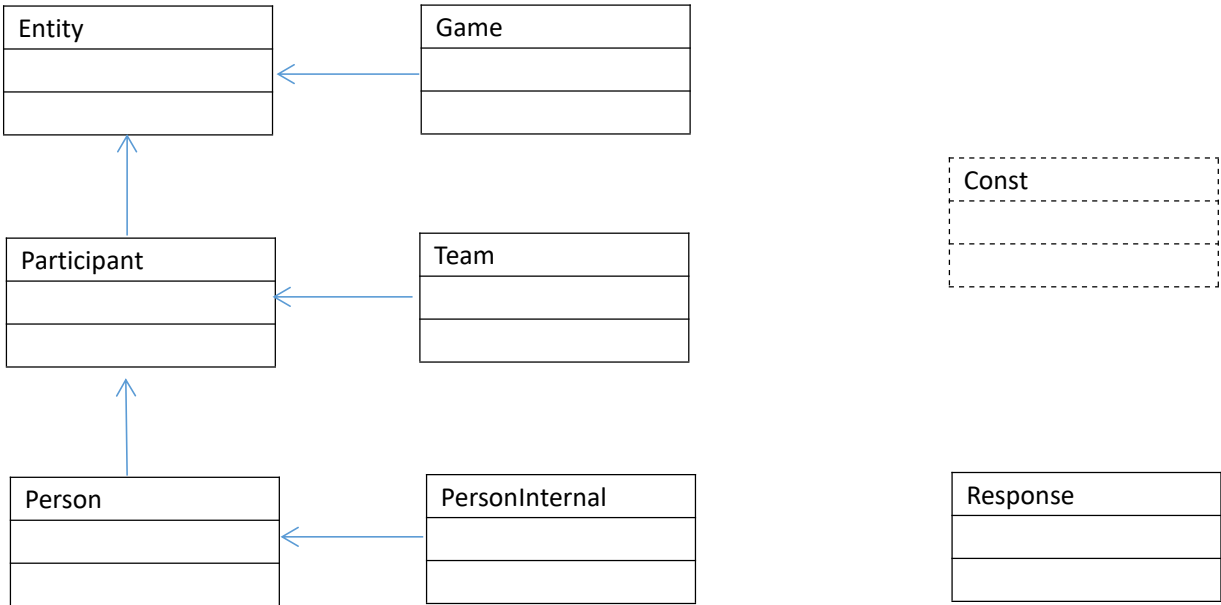
Extends: 
Implements: 

Class

Interface

2. Model

1) Overview

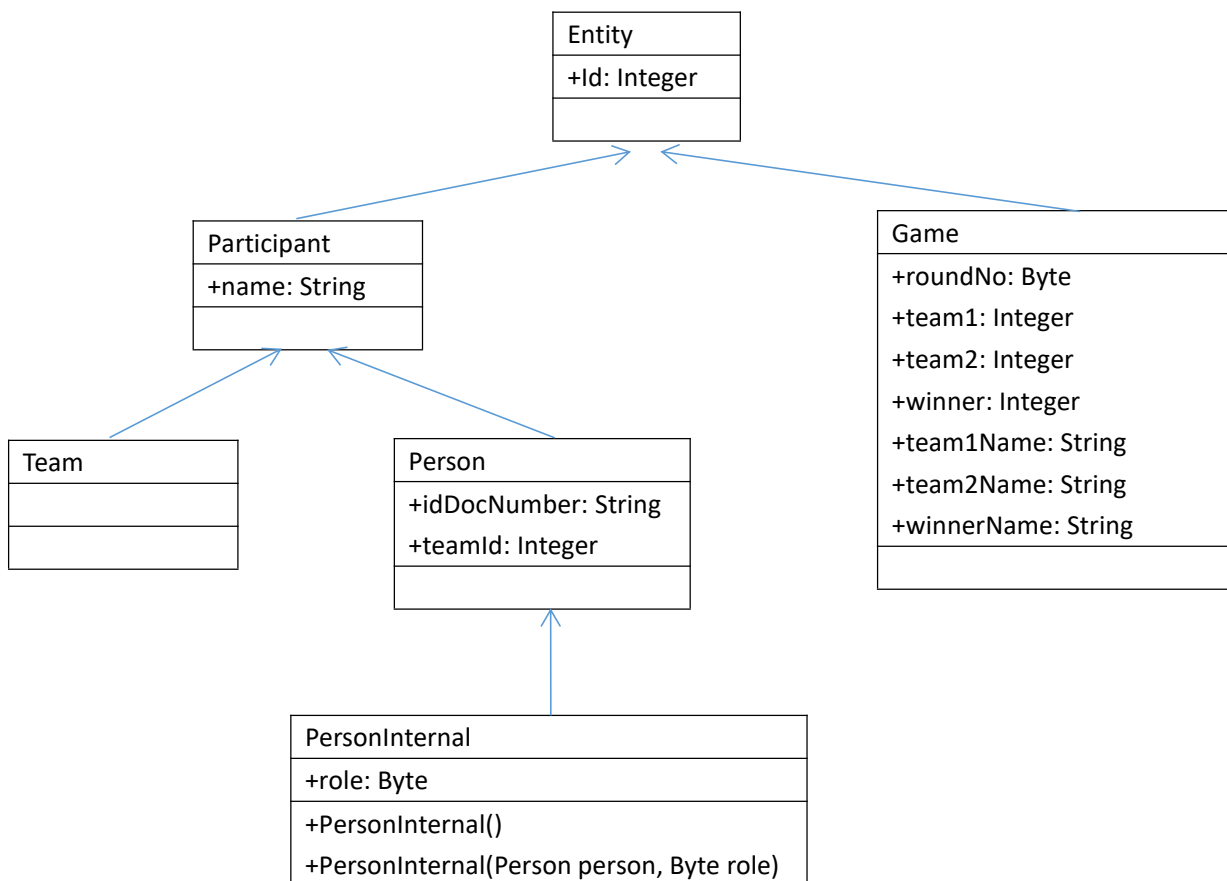


2) Response

Const(Interface)
Define Const used in the system:
Status code options
Status message options
Person roles
Max length of Strings
Max round no of game
Response format

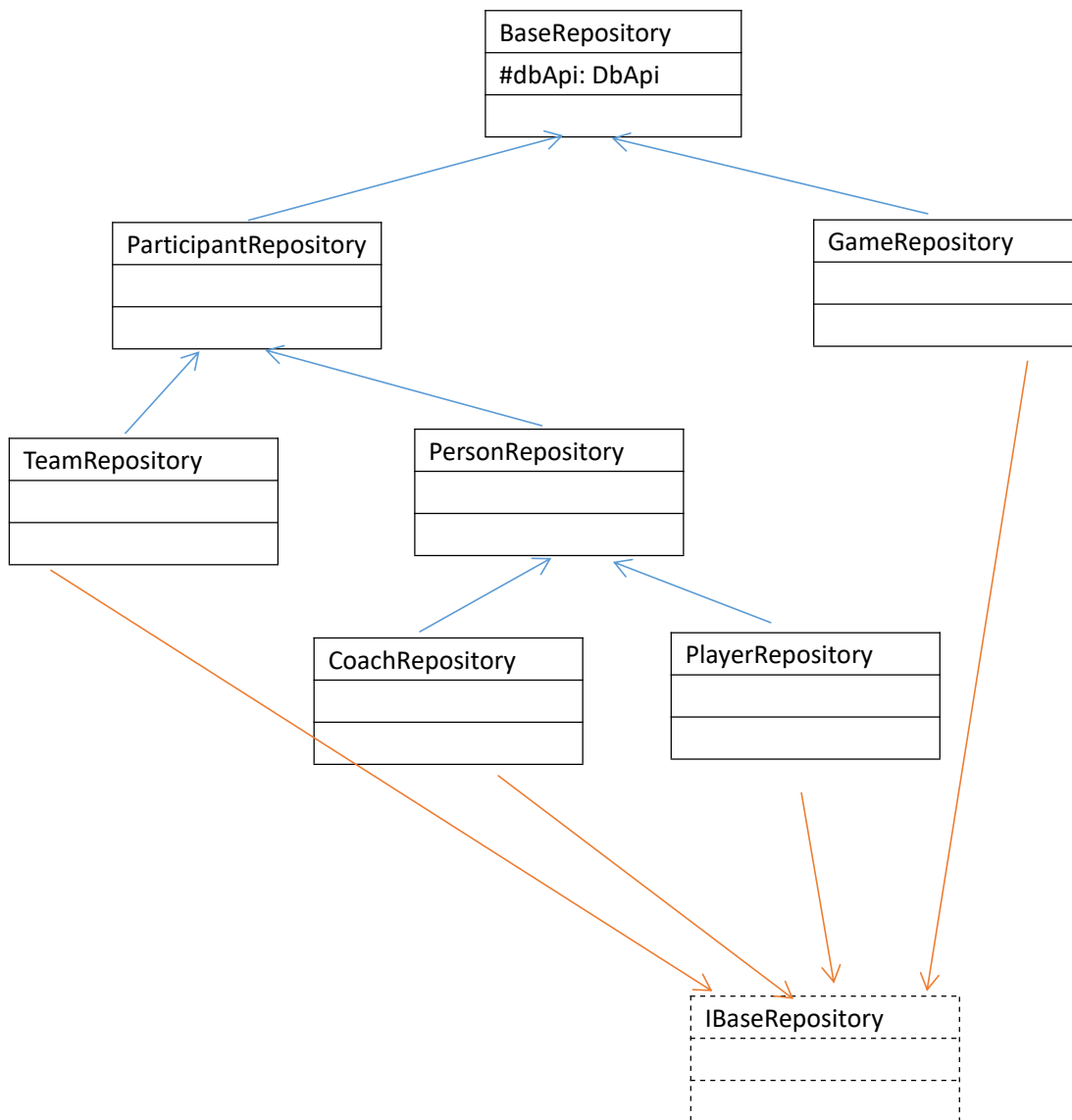
Response<T extends Entity>
+statusCode: int
+statusMsg: String
+entities: ArrayList<T>

3) Entities



3. Repository

1) Overview



2) DbApi

DbApi
<ul style="list-style-type: none">-mysqlHost: String-mysqlPort: String-mysqlDb: String-mysqlUsername: String-mysqlPassword: String-dataSource: HikariDataSource-lock: Lock
<ul style="list-style-type: none">+finalize()-initialize()~getConnection(): Connection

```

~readByColumn(String colName, String tableName, Class<? extends Entity> cls, Object colValue): Response
~readTable(String tableName, Class<? extends Entity> cls): Response
~readByFilters(String tableName, Class<? extends Entity> cls, String filters, ArrayList<Object> parameters): Response
~read(String sql, Class<? extends Entity> cls): Response
~read(String sql, ArrayList<Object> parameters, Class<? extends Entity> cls): Response
~getResultColumnFieldMap(Class cls, ResultSetMetaData resultSetMetaData): HashMap<String, Field>
~read(Connection connection, String sql, ArrayList<Object> parameters, Class<? extends Entity> cls): Response
~getColNames(Connection connection, String tableName): ArrayList<String>
~write(Optional<? extends Entity> optionalEntity, String tableName): Response
~write(Connection connection, Optional<? extends Entity> optionalEntity, String tableName): Response
~delete(Integer id, String tableName): Response
~delete(Connection connection, Integer id, String tableName): Response
~executeNonQuery(String sql, ArrayList<Object> parameters): Response
~executeNonQuery(Connection connection, String sql, ArrayList<Object> parameters): Response

```

3) IBaseRepository

```

IBaseRepository

+readAll(): Response<? extends Entity>
+readById(Integer id): Response<? extends Entity>
+delete(Integer id): Response

```

4) BaseRepository

```

BaseRepository
#dbApi: DbApi
+BaseRepository(DbApi dbApi)
#readById(Integer id, String tableName, Class cls): Response

```

5) GameRepository

```

GameRepository
#teamRepository: TeamRepository
~tableName: String
~cls: Class
~sqlReadAll: String
+GameRepository(DbApi dbApi, TeamRepository teamRepository)
+readAll(): Response
+readById(Integer id): Response
+read(Byte roundNo, Integer team1, Integer team2): Response
+setGameResult(Integer id, Integer winner): Response
+write(Game game): Response
+delete(Integer id): Response

```

6) ParticipantRepository

ParticipantRepository
+ParticipantRepository(DbApi dbApi) #readByName(String name, String tableName, Class cls): Response

7) TeamRepository

TeamRepository
- <u>tableName</u> : String - <u>cls</u> : Class - <u>sql4IsChildExists</u> : String
TeamRepository(DbApi dbApi) +readAll(): Response +readById(Integer id): Response +isChildExists(Integer id): Resonse +readByName(String name): Response +write(Team team): Response +delete(Integer id): Response

8) PersonRepository

PersonRepository
#teamRepository: TeamRepository - <u>tableName</u> : String - <u>clsOutput</u> : Class - <u>clsOutput1</u> : Class
PersonRepository(DbApi dbApi, TeamRepository teamRepository) #readByRole(Byte role): Response #readById(Integer id, Byte role): Response +readByIdDocNumber(String idDocNumber): Response #write(Person person, Byte role): Response #delete(Integer id, Byte role): Response

9) CoachRepository

CoachRepository
CoachRepository(DbApi dbApi, TeamRepository teamRepository) +readAll(): Response

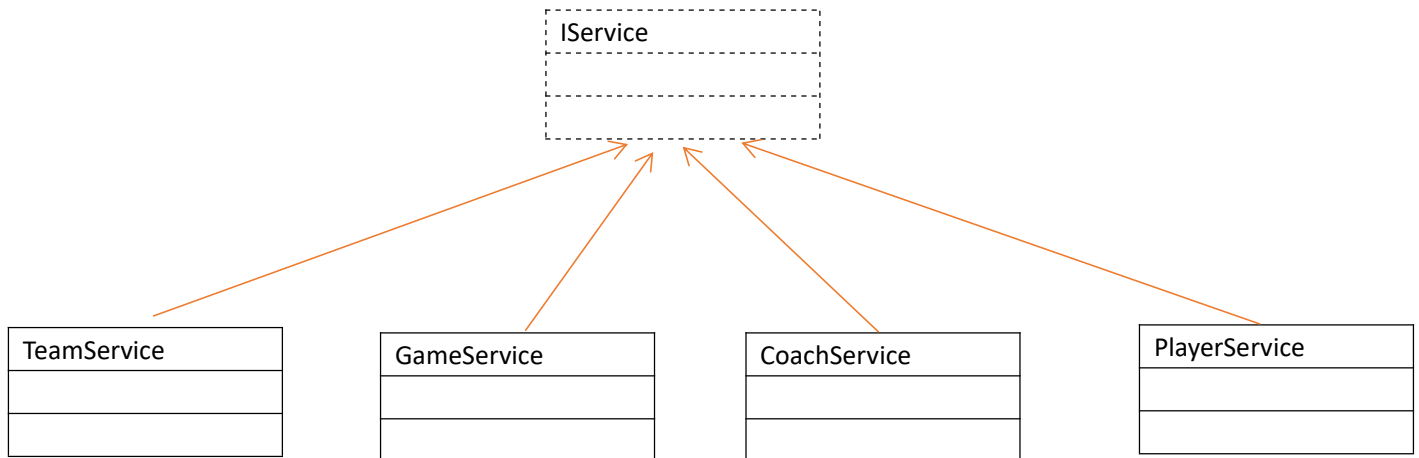
<pre>+readById(Integer id): Response +write(Person person): Response +delete(Integer id): Response</pre>
--

10)PlayerRepository

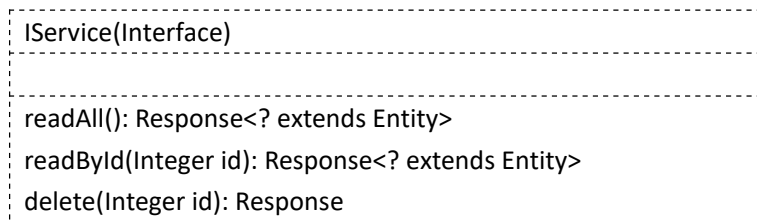
PlayerRepository
<pre>+PlayerRepository(DbApi dbApi, TeamRepository teamRepository) +readAll(): Response +readById(Integer id): Response +write(Person person): Response +delete(Integer id): Response</pre>

4. Service

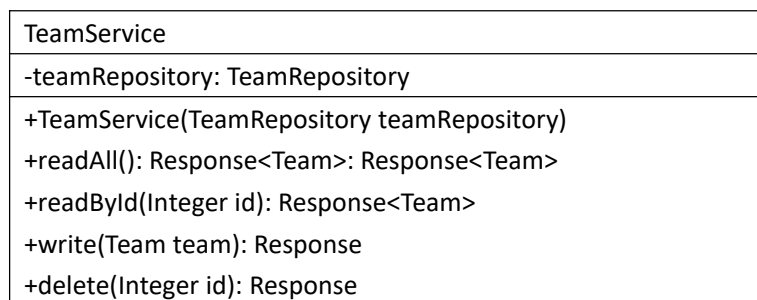
1) Overview



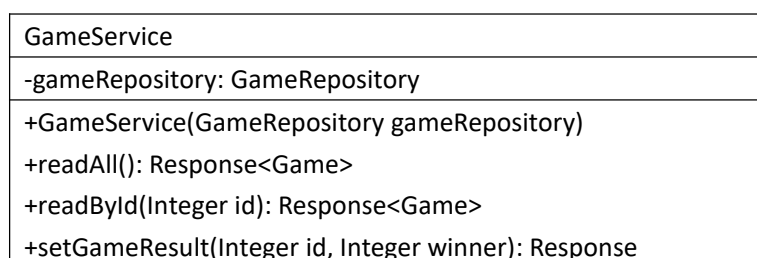
2) IService



3) TeamService



4) GameService



+write(Game game): Response +delete(Integer id): Response
--

5) CoachService

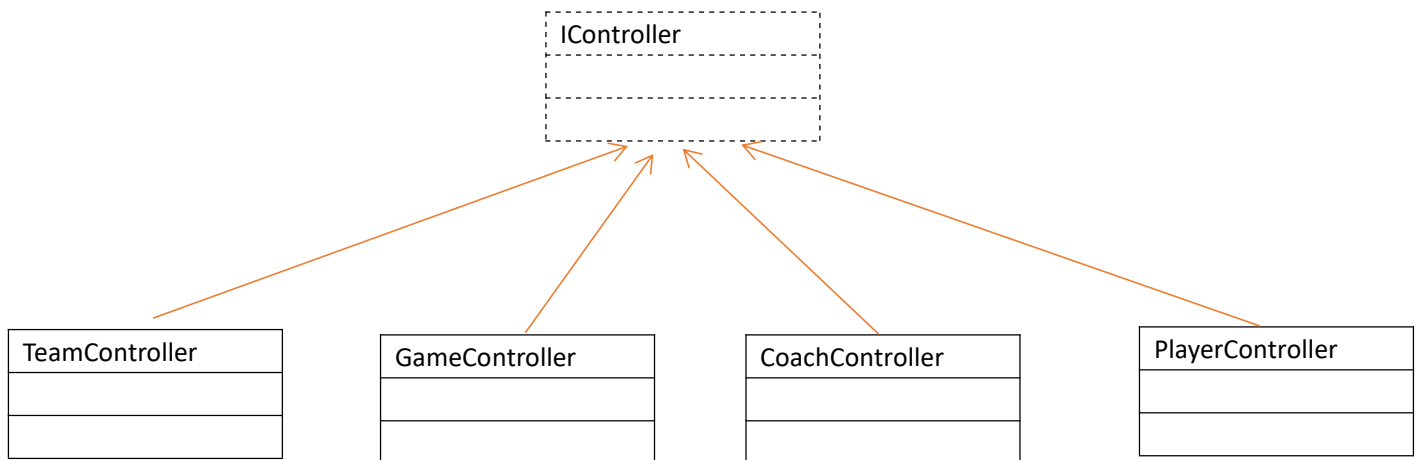
CoachService
-coachRepository: CoachRepository
+CoachService(CoachRepository coachRepository) +readAll(): Response<Person> +readById(Integer id): Response<Person> +write(Person person): Response +delete(Integer id): Response

6) PlayerService

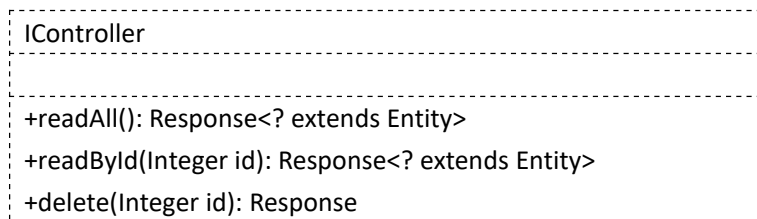
PlayerService
-playerRepository: PlayerRepository
+PlayerService(PlayerRepository playerRepository) +readAll(): Response<Person> +readById(Integer id): Response<Person> +write(Person person): Response +delete(Integer id): Response

5. Controller

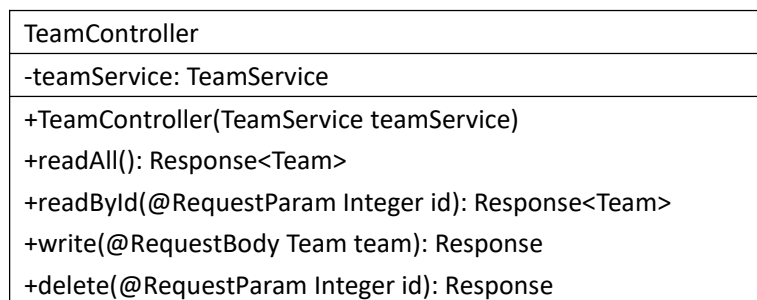
1) Overview



2) IController



3) TeamController



4) GameController



5) CoachController

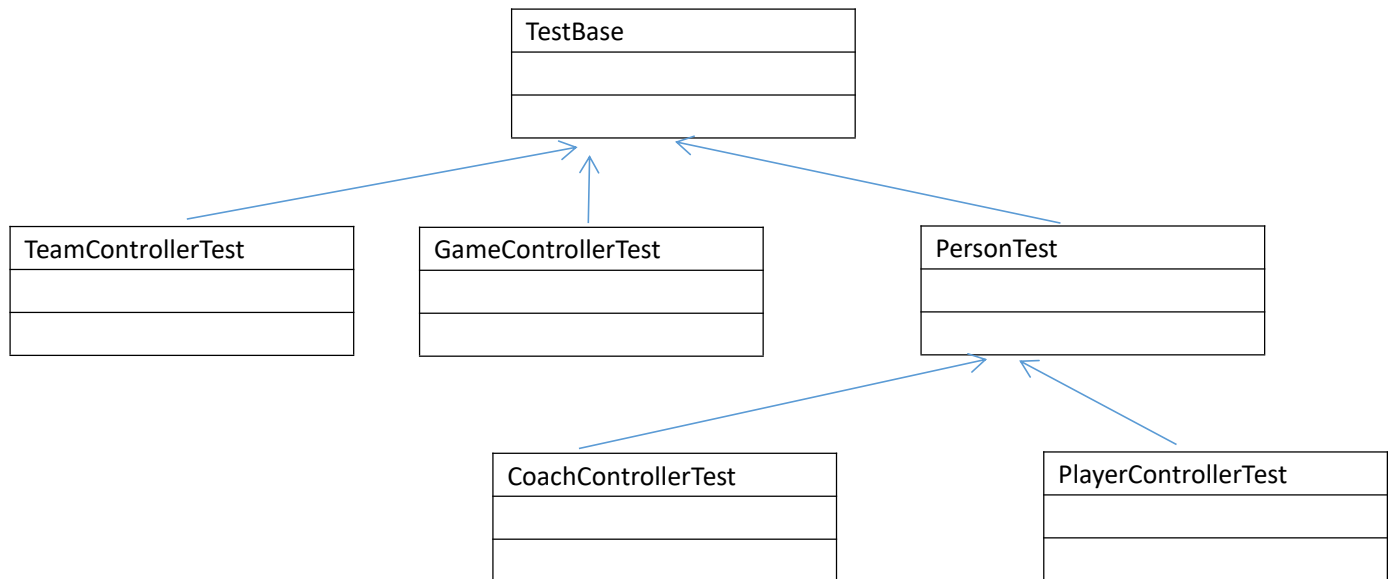
CoachController
-coachService: CoachService
+CoachController(CoachService coachService) +readAll(): Response<Person> +readById(@RequestParam Integer id): Response<Person> +write(@RequestBody Person coach): Response +delete(@RequestParam Integer id): Response

6) PlayerController

PlayerController
-playerService: PlayerService
+PlayerController(PlayerService playerService) +readAll(): Response<Person> +readById(@RequestParam Integer id): Response<Person> +write(@RequestBody Person coach): Response +delete(@RequestParam Integer id): Response

6. Test

1) Overview



2) TestBase

TestBase
<pre>#mockMvc: MockMvc #BASE_URL_TEAM: String #BASE_URL_GAME: String #BASE_URL_COACH: String #BASE_URL_PLAYER: String #gson: Gson #INVALID_STATUS: int #INVALID_ID: int -APPLICATION_JSON_UTF8: MediaType</pre>
<pre>#isStatusCodeValid(int statusCode): boolean #getStr(int length): String #readAll(String url) #readById(String url, Integer id): int #writeOnce(String url, Optional<? extends Entity> optionalEntity, int expectedResultCode, int unUnexpectedResultCode): int #writeATeam(): int #writeAGameForATeam(Integer team1, Integer team2): int #writeAGame(): int #writeAPersonForATeam(String baseUrl, Integer teamId): int #writeAPerson(String url): int #deleteAGame(int id, int expectedResultCode, int unUnexpectedResultCode) #deleteAPerson(String baseUrl, int id, int expectedResultCode, int unUnexpectedResultCode) #delete(String url, Integer id, int expectedResultCode, int unUnexpectedResultCode)</pre>

3) TeamControllerTest

TeamControllerTest
<code>~readAll()</code> <code>~readById()</code> <code>~writeOnce(Team team, int expectedResultCode, int unUnexpectedResultCode): int</code> <code>~write()</code> <code>~delete(int id, int expectedResultCode, int unUnexpectedResultCode)</code> <code>~delete()</code>

4) GameControllerTest

GameControllerTest
<code>~readAll()</code> <code>~readById()</code> <code>~writeOnce()</code> <code>~writeOnce(Game game, int expectedResultCode, int unUnexpectedResultCode): int</code> <code>~write()</code> <code>~delete()</code> <code>~setGameResult()</code> <code>-setGameResult(Integer id, Integer winner , int expectedResultCode, int unUnexpectedResultCode)</code>

5) PersonTest

PersonTest
<code>~writeACoach(): int</code> <code>~writeAPlayer(): int</code> <code>~writeOnce(String url, Person person, int expectedResultCode, int unUnexpectedResultCode): int</code> <code>~write(String url)</code>

6) CoachControllerTest

CoachControllerTest
<code>~readAll()</code> <code>~readById()</code> <code>~write()</code> <code>~delete(int id, int expectedResultCode, int unUnexpectedResultCode)</code> <code>~delete()</code>

7) PlayerControllerTest

PlayerControllerTest
<code>~readAll()</code> <code>~readById()</code> <code>~write()</code> <code>~delete(int id, int expectedResultCode, int unUnexpectedResultCode)</code> <code>~delete()</code>