Using the BattleArena API

BattleArena offers an API for developers to use if they wish to interface with the plugin.

The main entrypoint into BattleArena is the **BattleArenaApi** class, which can be accessed by running **BattleArenaApi.get()**.

The more important methods are documented below, but the full javadoc can be found here.

- **isInArena(Player)** - Returns whether the given Player is in an Arena
- **arena(String) / getArena(String)** - Returns the Arena from the given name, or null if not found
- **getArenas()** - Returns all the Arenas for the plugin
- **registerArena(Plugin, String, Class, Supplier)** - Registers the given Arena
- **getCompetitions(Arena)** - Returns all the existing Competitions for the Arena
- **getCompetitions(Arena, String)** - Returns all the Competitions for the given Arena and map name
- **findJoinableCompetition(List<Competition>, Player, PlayerRole)** - Finds a joinable competition for the given player with a player role

Useful Methods

A few methods that may be useful to plugin developers is also documented below.

Getting all players in an Arena

This shows how to get all players for a specific game across all maps and active competitions.

```
public List<Player> getAllPlayers(Arena arena) {
    List<Player> players = new ArrayList<>();

    for (Competition<?> competition : BattleArenaApi.get().getCompetitions(arena)) {
        // Ensure that the competition is occurring on this server
        if (!(competition instanceof LiveCompetition<?> liveCompetition)) {
            continue;
        }

        for (ArenaPlayer arenaPlayer : liveCompetition.getPlayers()) {
            players.add(arenaPlayer.getPlayer());
        }
```

```
    }

    return players;
}
```

Getting all maps

This shows how to get all maps that exist for an arena.

```
public List<? extends CompetitionMap> getMaps(Arena arena) {
    return BattleArenaApi.get().getMaps(arena);
}
```

Joining an arena

This shows how to make a player join an arena.

```
public void joinArena(Arena arena, Player player) {
    if (BattleArenaApi.get().isInArena(player)) {
        // Player is already in an arena
        return;
    }

    // Get all the competitions currently running
    List<Competition<?>> competitions = BattleArenaApi.get().getCompetitions(arena);

    // Search to see if there is a joinable competition for the player
    BattleArenaApi.get().findJoinableCompetition(competitions, player,
PlayerRole.PLAYING).thenAccept(result -> {
        JoinResult joinResult = result.result();

        // Success! They can join the competition
        if (joinResult.canJoin()) {
            result.competition().join(player, PlayerRole.PLAYING);
            return;
        }

        // The player cannot join the competition for some reason. This could be for any
reason:
        // - All competitions are full/running
        // - No maps have been set up
        // - No competitions are running
```

```
        // It is usually best practice to send the player a message as to why they cannot join
        joinResult.message().send(player);
    });
}
```

Revision #3
Created 2024-12-14 15:44:03 UTC by Redned
Updated 2024-12-14 16:03:47 UTC by Redned