

Package [dev.robocode.tankroyale.botapi](#)

Interface BotInfo.IBuilder

Enclosing class:

[BotInfo](#)

public static interface **BotInfo.IBuilder**

Builder interface for providing a builder for building [BotInfo](#) objects, and which supports method chaining.

Method Summary

All Methods Instance Methods Abstract Methods

Modifier and Type	Method	Description
BotInfo.IBuilder	addAuthor (java.lang.String author)	Adds an author of the bot.
BotInfo.IBuilder	addCountryCode (java.lang.String countryCode)	Adds a country code for the bot.
BotInfo.IBuilder	addGameType (java.lang.String gameType)	Adds a game type that this bot is capable of participating in.
BotInfo	build ()	Builds and returns the BotInfo instance based on the data set and added to this builder so far.

		this builder.
BotInfo.IBuilder	setAuthors (java.util.List<java.lang.String> authors)	Sets the names(s) of the author(s) of the bot.
BotInfo.IBuilder	setCountryCodes (java.util.List<java.lang.String> countryCodes)	Sets the country codes for the bot.
BotInfo.IBuilder	setDescription (java.lang.String description)	Sets a short description of the bot.
BotInfo.IBuilder	setGameTypes (java.util.Set<java.lang.String> gameTypes)	Sets the game types that this bot is capable of participating in.
BotInfo.IBuilder	setHomepage (java.lang.String homepage)	Sets a link to the homepage for the bot.
BotInfo.IBuilder	setInitialPosition (InitialPosition initialPosition)	Sets the initial position of this bot.
BotInfo.IBuilder	setName (java.lang.String name)	Sets the bot name.
BotInfo.IBuilder	setPlatform (java.lang.String platform)	Sets the name of the platform that this bot is build for.
BotInfo.IBuilder	setProgrammingLang (java.lang.String programmingLang)	Sets the name of the programming language

Method Detail

build

BotInfo build()

Builds and returns the BotInfo instance based on the data set and added to this builder so far. This method is typically the last method to call on the builder in order to extract the result of building.

Returns:
a BotInfo instance.

copy

BotInfo.IBuilder copy(BotInfo botInfo)

Copies all fields from a BotInfo instance into this builder.

Parameters:
botInfo - is the BotInfo instance to copy.

Returns:
this BotInfo.IBuilder instance provided for method chaining.

setName

BotInfo.IBuilder setName(java.lang.String name)

Sets the bot name. (required)

Note that the maximum length of the name is 30 characters.

Example of a name: "Rampage"

Parameters:

setVersion

```
BotInfo.IBuilder setVersion(java.lang.String version)
```

Sets the bot version. (required)

Note that the maximum length of the version is 20 characters.

Example of a version: "1.0"

Parameters:

version - is the version of the bot.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

setAuthors

```
BotInfo.IBuilder setAuthors(java.util.List<java.lang.String> authors)
```

Sets the names(s) of the author(s) of the bot. (required)

Note that the maximum length of an author name is 50 characters, and the maximum number of names is 5.

Example of the name of an author: "John Doe"

Parameters:

authors - is a list containing the names(s) of the author(s). A null removes all authors.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

See Also:

`addAuthor(java.lang.String)`

addAuthor

```
BotInfo.IBuilder addAuthor(java.lang.String author)
```

author - is the name of an author to add.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

See Also:

`setAuthors(java.util.List<java.lang.String>)`

setDescription

`BotInfo.IBuilder setDescription(java.lang.String description)`

Sets a short description of the bot. (optional)

Note that the maximum length of the description is 250 characters. Line-breaks (line-feed / new-line character) are supported, but only expect up to 3 lines to be displayed on a UI.

Example of a description:

```
"The rampage bot will try to ram bots that are very close.\nSneaks around the corners and shoot at the bots that come too near."
```

Parameters:

description - is a short description of the bot.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

setHomepage

`BotInfo.IBuilder setHomepage(java.lang.String homepage)`

Sets a link to the homepage for the bot. (optional)

Note that the maximum length of a link is 150 characters.

Example of a link: "https://fictive-homepage.net/Rampage"

Parameters:

homepage - is a link to a homepage for the bot.

Returns:

```
BotInfo.IBuilder setCountryCodes  
(java.util.List<java.lang.String> countryCodes)
```

Sets the country codes for the bot. (optional)

Note that the maximum length of each country code is 2 (alpha-2) from the ISO 3166 international standard, and the maximum number of country codes is 5.

Example of a country code: "dk"

Note that if no country code is specified, or the none of the country codes provided is valid, then the default a list containing a single country code will automatically be used containing the current locale country code. The current local country code will be extracted using `Locale.getDefault()`.

Parameters:

countryCodes - is a list containing the country codes. A null removes all country codes.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

See Also:

`addCountryCode(java.lang.String)`

addCountryCode

```
BotInfo.IBuilder addCountryCode(java.lang.String countryCode)
```

Adds a country code for the bot. (optional)

See `setCountryCodes(java.util.List<java.lang.String>)` for more details.

Parameters:

countryCode - is the country code to add.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

See Also:

`setCountryCodes(java.util.List<java.lang.String>)`

The standard game types [are listed here](#).

Note that more game types might be added in the future.

The `GameType` class contains the string for the current predefined game types, which can be used when setting the game types of this method.

Note that the maximum length of a game type is `20`, and the maximum number of game types is `10`.

Example of a game type: "classic"

Example of usage:

```
BotInfo.builder()  
    .setGameTypes(Set.of(GameType.CLASSIC, GameType.MELEE, "future-type"))  
    ...
```

Parameters:

`gameTypes` - is a set of game types that the bot is capable of participating in. A `null` removes all game types.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.

See Also:

[addGameType\(java.lang.String\)](#)

addGameType

```
BotInfo.IBuilder addGameType(java.lang.String gameType)
```

Adds a game type that this bot is capable of participating in. (required)

See [setGameTypes\(java.util.Set<java.lang.String>\)](#) for more details.

Example of usage:

```
BotInfo.builder()  
    .addGameType(GameType.CLASSIC)
```

SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#) DETAIL: FIELD | CONSTR | [METHOD](#)

`gameType` - is a game type that the bot is capable of participating in.

Returns:

this [BotInfo.IBuilder](#) instance provided for method chaining.

See Also:

`setGameTypes(java.util.Set<java.lang.String>)`

setPlatform

[BotInfo.IBuilder](#) `setPlatform(java.lang.String platform)`

Sets the name of the platform that this bot is build for. (optional)

Note that the maximum length of the name of the platform is 30.

If the platform is set to null or a blank string, then this default string will be used for this API:

Java Runtime Environment (JRE) [version]

Parameters:

`platform` - is the name of the platform that this bot is build for.

Returns:

this [BotInfo.IBuilder](#) instance provided for method chaining.

setProgrammingLang

[BotInfo.IBuilder](#) `setProgrammingLang(java.lang.String programmingLang)`

Sets the name of the programming language used for developing this bot. (optional)

Note that the maximum length of the name of the programming language is 30.

Parameters:

`programmingLang` - is the name of the programming language used for developing this bot.

Returns:

this [BotInfo.IBuilder](#) instance provided for method chaining.

Note that initial positions must be enabled/allowed with the game (server) in order to take effect.

Parameters:

`initialPosition` - is the initial position of this bot.

Returns:

this `BotInfo.IBuilder` instance provided for method chaining.