

**Package** dev.robocode.tankroyale.botapi

**Class BotInfo**

java.lang.Object  
dev.robocode.tankroyale.botapi.BotInfo

public final class **BotInfo**  
extends java.lang.Object

Bot info contains the properties of a bot.

**Nested Class Summary**

**Nested Classes**

Modifier and Type	Class	Description
static interface	<a href="#">BotInfo.IBuilder</a>	Builder interface for providing a builder for building <a href="#">BotInfo</a> objects, and which supports method chaining.

**Field Summary**

**Fields**

Modifier and Type	Field	Description
static int	<a href="#">MAX_AUTHOR_LENGTH</a>	Maximum number of characters accepted for an author name.
static int	<a href="#">MAX_DESCRIPTION_LENGTH</a>	Maximum number of characters accepted for the description.
static int	<a href="#">MAX_GAME_TYPE_LENGTH</a>	Maximum number of characters accepted for a game type.
static int	<a href="#">MAX_HOMEPAGE_LENGTH</a>	Maximum number of characters accepted for the link to the homepage.
static int	<a href="#">MAX_NAME_LENGTH</a>	Maximum number of characters accepted for the name.
static int	<a href="#">MAX_NUMBER_OF_AUTHORS</a>	Maximum number of authors accepted.
static int	<a href="#">MAX_NUMBER_OF_COUNTRY_CODES</a>	Maximum number of country codes accepted.
static int	<a href="#">MAX_NUMBER_OF_GAME_TYPES</a>	Maximum number of game types accepted.
static int	<a href="#">MAX_PLATFORM_LENGTH</a>	Maximum number of characters accepted for the platform name.
static int	<a href="#">MAX_PROGRAMMING_LANG_LENGTH</a>	Maximum number of characters accepted for the programming language name.
static int	<a href="#">MAX_VERSION_LENGTH</a>	Maximum number of characters accepted for the version.

Constructor	Description
<b>BotInfo</b> (java.lang.String name, java.lang.String version, java.util.List<java.lang.String> authors, java.lang.String description, java.lang.String homepage, java.util.List<java.lang.String> countryCodes, java.util.Collection<java.lang.String> gameTypes, java.lang.String platform, java.lang.String programmingLang, <b>InitialPosition</b> initialPosition)	<p>Initializes a new instance of the BotInfo class.</p> <p>Note that the recommended method for creating a BotInfo class is to use the <b>BotInfo.IBuilder</b> interface provided with the static <b>builder()</b> method.</p>

Method Summary

All Methods	Static Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description	
static <b>BotInfo.IBuilder</b>	<b>builder()</b>	Returns a builder for a convenient way of building a <b>BotInfo</b> object using the <b>builder</b> pattern.  Example of use:	
static <b>BotInfo</b>	<b>fromFile</b> (java.lang.String filename)	Reads the bot info from a local file on a file system. The file is assumed to be in JSON format.  See the <b>fromInputStream(java.io.InputStream)</b> to see the required JSON format for the file.	
static <b>BotInfo</b>	<b>fromInputStream</b> (java.io.InputStream inputStream)	Reads the bot info from an input stream. The file is assumed to be in JSON format.  Example file in JSON format:	
static <b>BotInfo</b>	<b>fromResourceFile</b> (java.lang.String filename)	Reads the bot info from a resource file, e.g.	
java.util.List<java.lang.String>	<b>getAuthors()</b>	Returns the list of authors of the bot, e.g., "John Doe (johndoe@somewhere.io)".	
java.util.List<java.lang.String>	<b>getCountryCodes()</b>	Returns a list of country code(s) defined by <b>ISO 3166-1 alpha-2</b> , e.g.	
java.lang.String	<b>getDescription()</b>	Returns a short description of the bot, preferably a one-liner. This field is optional.	
java.util.Set<java.lang.String>	<b>getGameTypes()</b>	Returns the game type(s) accepted by the bot, e.g., "classic", "melee", "1v1".	
java.lang.String	<b>getHomepage()</b>	Returns the URL of a web page for the bot. This field is optional.	

ALL CLASSES

SEARCH:

SUMMARY: NESTED | FIELD | CONSTR | METHOD      DETAIL: FIELD | CONSTR | METHOD

java.lang.String	<a href="#">getName()</a>	Returns the name, e.g., "MyBot".
java.lang.String	<a href="#">getPlatform()</a>	Returns the platform used for running the bot, e.g., "Java Runtime Environment (JRE) 11". This field is optional.
java.lang.String	<a href="#">getProgrammingLang()</a>	Returns the programming language used for developing the bot, e.g., "Java 11" or "Kotlin 1.7.20". This field is optional.
java.lang.String	<a href="#">getVersion()</a>	Returns the version, e.g., "1.0".

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

MAX\_NAME\_LENGTH

public static final int MAX\_NAME\_LENGTH

Maximum number of characters accepted for the name.

See Also:

[Constant Field Values](#)

MAX\_VERSION\_LENGTH

public static final int MAX\_VERSION\_LENGTH

Maximum number of characters accepted for the version.

See Also:

[Constant Field Values](#)

MAX\_AUTHOR\_LENGTH

public static final int MAX\_AUTHOR\_LENGTH

Maximum number of characters accepted for an author name.

See Also:

[Constant Field Values](#)

SUMMARY: NESTED | FIELD | CONSTR | METHOD     DETAIL: FIELD | CONSTR | METHOD  
Maximum number of characters accepted for the description.

**See Also:**  
[Constant Field Values](#)

**MAX\_HOMEPAGE\_LENGTH**

```
public static final int MAX_HOMEPAGE_LENGTH
```

Maximum number of characters accepted for the link to the homepage.

**See Also:**  
[Constant Field Values](#)

**MAX\_GAME\_TYPE\_LENGTH**

```
public static final int MAX_GAME_TYPE_LENGTH
```

Maximum number of characters accepted for a game type.

**See Also:**  
[Constant Field Values](#)

**MAX\_PLATFORM\_LENGTH**

```
public static final int MAX_PLATFORM_LENGTH
```

Maximum number of characters accepted for the platform name.

**See Also:**  
[Constant Field Values](#)

**MAX\_PROGRAMMING\_LANG\_LENGTH**

```
public static final int MAX_PROGRAMMING_LANG_LENGTH
```

Maximum number of characters accepted for the programming language name.

**See Also:**  
[Constant Field Values](#)

**MAX\_NUMBER\_OF\_AUTHORS**

```
public static final int MAX_NUMBER_OF_AUTHORS
```

Maximum number of authors accepted.

**See Also:**

```
public static final int MAX_NUMBER_OF_COUNTRY_CODES
```

Maximum number of country codes accepted.

**See Also:**

[Constant Field Values](#)

**MAX\_NUMBER\_OF\_GAME\_TYPES**

```
public static final int MAX_NUMBER_OF_GAME_TYPES
```

Maximum number of game types accepted.

**See Also:**

[Constant Field Values](#)

**Constructor Detail**

**BotInfo**

```
public BotInfo(java.lang.String name,
               java.lang.String version,
               java.util.List<java.lang.String> authors,
               java.lang.String description,
               java.lang.String homepage,
               java.util.List<java.lang.String> countryCodes,
               java.util.Collection<java.lang.String> gameTypes,
               java.lang.String platform,
               java.lang.String programmingLang,
               InitialPosition initialPosition)
```

Initializes a new instance of the BotInfo class.

Note that the recommended method for creating a BotInfo class is to use the [BotInfo.IBuilder](#) interface provided with the static [builder\(\)](#) method.

**Parameters:**

name - is the name of the bot (required).

version - is the version of the bot (required).

authors - is the author(s) of the bot (required).

description - is a short description of the bot (optional).

homepage - is the link to a homepage for the bot (optional).

countryCodes - is the country code(s) for the bot (optional).

gameTypes - is the game types that this bot can handle (optional).

platform - is the platform used for running the bot (optional).

programmingLang - is the programming language used for developing the bot (optional).

**Method Detail****builder**

```
public static BotInfo.IBuilder builder()
```

Returns a builder for a convenient way of building a `BotInfo` object using the [builder pattern](#).

Example of use:

```
BotInfo botInfo = BotInfo.builder()
    .setName("Rampage")
    .setVersion("1.0")
    .addAuthor("John Doh")
    .setGameTypes(List.of(GameType.CLASSIC, GameType.MELEE))
    .build();
```

**Returns:**

a builder for building a `BotInfo` object.

**getName**

```
public java.lang.String getName()
```

Returns the name, e.g., "MyBot". This field must always be provided with the bot info.

**Returns:**

The name of the bot.

**getVersion**

```
public java.lang.String getVersion()
```

Returns the version, e.g., "1.0". This field must always be provided with the bot info.

**Returns:**

The version of the bot.

**getAuthors**

```
public java.util.List<java.lang.String> getAuthors()
```

Returns the list of authors of the bot, e.g., "John Doe (johndoe@somewhere.io)". At least one author must be provided.

**Returns:**

The name(s) of the author(s) of the bot.

Returns a short description of the bot, preferably a one-liner.  
This field is optional.

**Returns:**

a short description of the bot.

**getHomepage**

```
public java.lang.String getHomepage()
```

Returns the URL of a web page for the bot.  
This field is optional.

**Returns:**

The URL of a web page for the bot.

**getCountryCodes**

```
public java.util.List<java.lang.String> getCountryCodes()
```

Returns a list of country code(s) defined by [ISO 3166-1 alpha-2](#), e.g. { "us" } This field is optional. If no country codes are provided, the locale of the system is being used instead.

**Returns:**

The country code(s) for the bot.

**getGameTypes**

```
public java.util.Set<java.lang.String> getGameTypes()
```

Returns the game type(s) accepted by the bot, e.g., "classic", "melee", "1v1". At least one game type must be provided to indicate the type(s) of games that this bot can participate in. The game types define which game types the bot can participate in. See [GameType](#) for using predefined game type.

**Returns:**

The game type(s) that this bot can handle.

**getPlatform**

```
public java.lang.String getPlatform()
```

Returns the platform used for running the bot, e.g., "Java Runtime Environment (JRE) 11".  
This field is optional.

**Returns:**

The platform used for running the bot.

Returns the programming language used for developing the bot, e.g., java 11 or Kotlin 1.7.20 .  
This field is optional.

**Returns:**

The programming language used for developing the bot.

**getInitialPosition**

```
public InitialPosition getInitialPosition()
```

Returns the initial starting position used for debugging only, which must be enabled at the server.  
This field is optional.

**Returns:**

The initial starting position used for debugging only.

**fromResourceFile**

```
public static BotInfo fromResourceFile(java.lang.String filename)
```

Reads the bot info from a resource file, e.g. when the file is located in a jar file or resource path in IDE.  
The file is assumed to be in JSON format.

See the `fromInputStream(java.io.InputStream)` to see the required JSON format for the file.

**Parameters:**

filename - is the filename of the file containing bot properties.

**Returns:**

A BotInfo instance containing the bot properties read from the file.

**Throws:**

`BotException` - if the resource file could not be read, or if some field read from the file is invalid.

**See Also:**

`fromFile(java.lang.String)`, `fromInputStream(java.io.InputStream)`

**fromFile**

```
public static BotInfo fromFile(java.lang.String filename)
```

Reads the bot info from a local file on a file system.  
The file is assumed to be in JSON format.

See the `fromInputStream(java.io.InputStream)` to see the required JSON format for the file.

**Parameters:**

filename - is the filename of the file containing bot properties.

**Returns:**

A BotInfo instance containing the bot properties read from the file.

**Throws:**

`BotException` - if the file could not be read, or if some field read from the file is invalid.



**fromInputStream**

```
public static BotInfo fromInputStream(java.io.InputStream inputStream)
```

Reads the bot info from an input stream.  
The file is assumed to be in JSON format.

Example file in JSON format:

```
{
  "name": "MyBot",
  "version": "1.0",
  "authors": "John Doe",
  "description": "Short description",
  "homepage": "https://somewhere.net/MyBot",
  "countryCodes": "us",
  "gameTypes": "classic, melee, 1v1",
  "platform": "JVM",
  "programmingLang": "Java 11",
  "initialPosition": "50,50, 90"
}
```

Note that these fields are required as these are used to identify the bot:

- name
- version
- authors

These value can take multiple values separated by a comma:

- authors, e.g. "John Doe, Jane Doe"
- countryCodes, e.g. "se, no, dk"
- gameTypes, e.g. "classic, melee, 1v1"

The `initialPosition` variable is optional and should *only* be used for debugging.

The `gameTypes` is optional, but can be used to limit which game types the bot is capable of participating in.

**Parameters:**

`inputStream` - is the input stream providing the bot properties.

**Returns:**

A `BotInfo` instance containing the bot properties read from the stream.

**Throws:**

`BotException` - if some fields read from the stream is invalid.

**See Also:**

`fromFile(java.lang.String)`, `fromResourceFile(java.lang.String)`