

Namespace ArgumentAPI

Classes

[APIControlUnit](#)

The loadbearing class, with which the user is able to communicate when defining options and their parameters, as well as when parsing the values obtained from these options during runtime.

[Enum](#)

Represents the Enum type that can be specified as an argument type. To facilitate working with it, an implicit cast is defined, so that the Enum class acts as a List<>.

[EnumParameter](#)

Encapsulates enum parameters for options.

[EnumParameterBuilder](#)

Provides method overrides for Enum type parameters.

[Float](#)

Represents the Float (float) type that can be specified as an argument type. It is encapsulated in a class because float is a value type, which does not work with the "dynamic" keyword. An implicit cast is defined so that the Float class acts as a float.

[FloatParameter](#)

Encapsulates float parameters for options.

[FloatParameterBuilder](#)

Provides method overrides for float type parameters.

[Int](#)

Represents the Int (int) type that can be specified as an argument type. It is encapsulated in a class because int is a value type, which does not work with the "dynamic" keyword. An implicit cast is defined so that the Int class acts as an int.

[IntParameter](#)

Encapsulates int parameters for options.

[IntParameterBuilder](#)

Provides method overrides for int type parameters.

[Option](#)

Houses the Option definitions.

[OptionBuilder](#)

Builder pattern class; used to provide a fluid option defining experience to the user, while not exposing them to the Option objects.

[Parameter](#)

Parent class for all types of parameters that can be assigned to options.

[ParameterBuilder](#)

Builder pattern class used to provide a fluid parameter defining experience to the user, while not exposing them to the Parameter objects.

[StringParameter](#)

Encapsulates string parameters for options.

[StringParameterBuilder](#)

Provides method overrides for string type parameters.

Class APIControlUnit

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








The loadbearing class, with which the user is able to communicate when defining options and their parameters, as well as when parsing the values obtained from these options during runtime.

```
public class APIControlUnit
```

Inheritance

[object](#)  ← APIControlUnit

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Methods

GetOptionNames()

Allows the user to access the names of all options that have been set in the command line, it will always return the name of the long option, if it has both a long and short variants, and only return the short name if that is the only variant of that command.

```
public List<string> GetOptionNames()
```

Returns

[List](#)  <[string](#)  >

A list of strings representing the names of the options.

GetOptionParameter(string)

Allows the user to retrieve the Parameter object for a specified option. The user is assumed to know what type of parameters they expect from each of the options.

```
public Parameter GetOptionParameter(string optionName)
```

Parameters

`optionName` [string](#) 

The name of the option for which the parameter is requested.

Returns

[Parameter](#)

A Parameter object corresponding to the specified option name.

ParseArgs(string[])

Parses all the arguments in the command line, and prepares the APIControlUnit instance for queries about the parameters through its methods.

```
public void ParseArgs(string[] args)
```

Parameters

`args` [string](#)  []

The array of command-line arguments to be parsed.

ParseFile(string)

Reads in the options that are specified in a configuration file, and stores them in the APIControlUnit instance. This file will be in a CSV format, with parameters being enclosed in {}. Since defining arguments in both a file and code is possible sequentially, if there is a problem with the file, the user will be issued a warning, but the program will continue.

```
public void ParseFile(string filename)
```

Parameters

filename [string](#)

The name of the configuration file to be parsed.

Class Enum

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Represents the Enum type that can be specified as an argument type. To facilitate working with it, an implicit cast is defined, so that the Enum class acts as a List<>.

```
public class Enum
```

Inheritance

[object](#)  ← Enum

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

Value

Gets the value of the Enum as a list of strings.

```
public List<string> Value { get; }
```

Property Value

[List](#)  <[string](#)  >

Operators

implicit operator List<string>(Enum)

Implicit cast to convert an Enum object to a List<string>.

```
public static implicit operator List<string>(Enum enumerator)
```

Parameters

enumerator [Enum](#)

The Enum object to convert.

Returns

[List](#) <[string](#)>

Class EnumParameter

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll

Encapsulates enum parameters for options.

```
public class EnumParameter : Parameter
```

Inheritance

[object](#)  ← [Parameter](#) ← EnumParameter

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

Value()

Returns the value of the Enum parameter.

```
public override Enum Value()
```

Returns

[Enum](#)

Class EnumParameterBuilder

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Provides method overrides for Enum type parameters.

```
public class EnumParameterBuilder : ParameterBuilder
```

Inheritance

[object](#)  ← [ParameterBuilder](#) ← EnumParameterBuilder

Inherited Members

[ParameterBuilder.IsRequired\(\)](#), [ParameterBuilder.IsString\(\)](#), [ParameterBuilder.IsEnum\(\)](#), [ParameterBuilder.IsInt\(\)](#), [ParameterBuilder.IsFloat\(\)](#), [ParameterBuilder.WithLambdaValidation\(\)](#), [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

GetParameter()

Returns the built EnumParameter object.

```
public override EnumParameter GetParameter()
```

Returns

[EnumParameter](#)

WithDefaultValue(string)

Sets the default value for the Enum parameter. A warning is thrown if the given string is not within the defined Enum values.

```
public EnumParameterBuilder WithDefaultValue(string value)
```

Parameters

value [string](#)

Returns

[EnumParameterBuilder](#)

WithEnumOption(string)

Adds an Enum value to the accepted values.

```
public EnumParameterBuilder WithEnumOption(string option)
```

Parameters

option [string](#)

Returns

[EnumParameterBuilder](#)

Class Float

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Represents the Float (float) type that can be specified as an argument type. It is encapsulated in a class because float is a value type, which does not work with the "dynamic" keyword. An implicit cast is defined so that the Float class acts as a float.

```
public class Float
```

Inheritance

[object](#)  ← Float

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

Value

Gets or sets the value of the Float.

```
public float Value { get; set; }
```

Property Value

[float](#) 

Operators

implicit operator float(Float)

Implicit cast to convert a Float object to a float.

```
public static implicit operator float(Float floating)
```

Parameters

floating [Float](#)

The Float object to convert.

Returns

[float](#)

Class FloatParameter

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Encapsulates float parameters for options.

```
public class FloatParameter : Parameter
```

Inheritance

[object](#)  ← [Parameter](#) ← FloatParameter

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

Value()

Returns the value of the float parameter.

```
public override Float Value()
```

Returns

[Float](#)

Class FloatParameterBuilder

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Provides method overrides for float type parameters.

```
public class FloatParameterBuilder : ParameterBuilder
```

Inheritance

[object](#)  ← [ParameterBuilder](#) ← FloatParameterBuilder

Inherited Members

[ParameterBuilder.IsRequired\(\)](#), [ParameterBuilder.IsString\(\)](#), [ParameterBuilder.IsEnum\(\)](#), [ParameterBuilder.IsInt\(\)](#), [ParameterBuilder.IsFloat\(\)](#), [ParameterBuilder.WithLambdaValidation\(\)](#), [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

GetParameter()

Returns the built FloatParameter object.

```
public override FloatParameter GetParameter()
```

Returns

[FloatParameter](#)

WithDefaultValue(float)

Sets the default value for the float parameter. A warning is thrown if the provided float value is outside the defined bounds or does not satisfy the lambda validation function.

```
public FloatParameterBuilder WithDefaultValue(float value)
```

Parameters

value [float](#)

Returns

[FloatParameterBuilder](#)

WithLambdaValidation(Func<float, bool>?)

Accepts a lambda function for float validation.

```
public FloatParameterBuilder WithLambdaValidation(Func<float, bool>? validation = null)
```

Parameters

validation [Func](#) <[float](#), [bool](#)>

Returns

[FloatParameterBuilder](#)

WithLowerBound(float)

Allows the user to set a lower bound for the valid values of the float parameter. This value IS included in the valid range.

```
public FloatParameterBuilder WithLowerBound(float bound)
```

Parameters

bound [float](#)

Returns

[FloatParameterBuilder](#)

WithUpperBound(float)

Allows the user to set an upper bound for the valid values of the float parameter. This value IS NOT included in the valid range.

```
public FloatParameterBuilder WithUpperBound(float bound)
```

Parameters

bound [float](#)

Returns

[FloatParameterBuilder](#)

Class Int

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Represents the Int (int) type that can be specified as an argument type. It is encapsulated in a class because int is a value type, which does not work with the "dynamic" keyword. An implicit cast is defined so that the Int class acts as an int.

```
public class Int
```

Inheritance

[object](#)  ← Int

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Properties

Value

Gets or sets the value of the Int.

```
public int Value { get; set; }
```

Property Value

[int](#) 

Operators

implicit operator int(Int)

Implicit cast to convert an Int object to an int.

```
public static implicit operator int(Int integer)
```

Parameters

integer [Int](#)

The Int object to convert.

Returns

[int](#)

Class IntParameter

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Encapsulates int parameters for options.

```
public class IntParameter : Parameter
```

Inheritance

[object](#)  ← [Parameter](#) ← IntParameter

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

Value()

Returns the value of the int parameter.

```
public override Int Value()
```

Returns

[Int](#)

Class IntParameterBuilder

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Provides method overrides for int type parameters.

```
public class IntParameterBuilder : ParameterBuilder
```

Inheritance

[object](#)  ← [ParameterBuilder](#) ← IntParameterBuilder

Inherited Members

[ParameterBuilder.IsRequired\(\)](#), [ParameterBuilder.IsString\(\)](#), [ParameterBuilder.IsEnum\(\)](#), [ParameterBuilder.IsInt\(\)](#), [ParameterBuilder.IsFloat\(\)](#), [ParameterBuilder.WithLambdaValidation\(\)](#), [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

GetParameter()

Returns the built IntParameter object.

```
public override IntParameter GetParameter()
```

Returns

[IntParameter](#)

WithDefaultValue(int)

Sets the default value for the int parameter. A warning is thrown if the provided value is outside the defined bounds or does not satisfy the lambda validation function.

```
public IntParameterBuilder WithDefaultValue(int value)
```

Parameters

value [int](#)

Returns

[IntParameterBuilder](#)

WithLambdaValidation(Func<int, bool>?)

Accepts a lambda function for int validation.

```
public IntParameterBuilder WithLambdaValidation(Func<int, bool>? validation = null)
```

Parameters

validation [Func](#) <[int](#), [bool](#)>

Returns

[IntParameterBuilder](#)

WithLowerBound(int)

Allows the user to set a lower bound for the valid values of the int parameter. This value IS included in the valid range.

```
public IntParameterBuilder WithLowerBound(int bound)
```

Parameters

bound [int](#)

Returns

[IntParameterBuilder](#)

WithUpperBound(int)

Allows the user to set an upper bound for the valid values of the int parameter. This value IS NOT included in the valid range.

```
public IntParameterBuilder WithUpperBound(int bound)
```

Parameters

bound [int](#)

Returns

[IntParameterBuilder](#)

Class Option

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Houses the Option definitions.

```
public class Option
```

Inheritance

[object](#)  ← Option

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Constructors

Option(string)

Parametrized constructor which takes the option name.

```
public Option(string name)
```

Parameters

name [string](#) 

The name of the option.

Class OptionBuilder


Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Builder pattern class; used to provide a fluid option defining experience to the user, while not exposing them to the Option objects.

```
public class OptionBuilder
```

Inheritance

[object](#)  ← OptionBuilder

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Constructors

OptionBuilder(string)

Parametrized constructor, takes the name of the option.

```
public OptionBuilder(string name)
```

Parameters

name [string](#) 

The name of the option.

Methods

IsLong()

Sets the option as a long option. Throws an error if the name is already taken. The option will also not be accepted without this, or the IsShort method.


```
public OptionBuilder IsLong()
```

Returns

[OptionBuilder](#)

IsRequired()

Sets the option as required, so if it is not set when the program runs, the user will be issued a warning.

```
public OptionBuilder IsRequired()
```

Returns

[OptionBuilder](#)

IsShort()

Sets the option as a short option. Throws an error if the name is not 1 character long, or if the name is already taken. The option will also not be accepted without this, or the IsLong method.

```
public OptionBuilder IsShort()
```

Returns

[OptionBuilder](#)

SetOption(APIControlUnit)

Sets the option up in the APIControlUnit object, so it is ready for argument parsing.

```
public OptionBuilder SetOption(APIControlUnit control)
```

Parameters

control [APIControlUnit](#)

The APIControlUnit object to set the option in.

Returns

[OptionBuilder](#)

WithAlias(string)

Allows the user to set an alias for the option. An error is thrown if the name is already taken.

```
public OptionBuilder WithAlias(string alias)
```

Parameters

alias [string](#) 

The alias for the option.

Returns

[OptionBuilder](#)

WithHelp(string)

Allows the user to define the help text for the option.

```
public OptionBuilder WithHelp(string help)
```

Parameters

help [string](#) 

The help text for the option.

Returns

[OptionBuilder](#)

WithPair(string)

Allows the user to set a paired name for the option. If it is a long option, the pair must be a short option, and vice versa. An error will be thrown if the name is already taken, the name doesn't conform to the type of option it should be, or the method is used before the `IsShort` or `IsLong` methods.

```
public OptionBuilder WithPair(string name)
```

Parameters

name [string](#) 

The paired name for the option.

Returns

[OptionBuilder](#)

WithParameter(Parameter)

Allows the user to set a parameter for the option in the form of a built `Parameter` object.

```
public OptionBuilder WithParameter(Parameter parameter)
```

Parameters

parameter [Parameter](#)

The parameter for the option.

Returns

[OptionBuilder](#)

Class Parameter


Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll

Parent class for all types of parameters that can be assigned to options.

```
public class Parameter
```








Inheritance

[object](#)  ← Parameter

Derived

[EnumParameter](#), [FloatParameter](#), [IntParameter](#), [StringParameter](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

Value()

Virtual method to be overloaded in child classes to return the value of the parameter.

```
public virtual dynamic Value()
```

Returns

dynamic

Class ParameterBuilder


Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll

Builder pattern class used to provide a fluid parameter defining experience to the user, while not exposing them to the Parameter objects.

```
public class ParameterBuilder
```








Inheritance

[object](#)  ← ParameterBuilder

Derived

[EnumParameterBuilder](#), [FloatParameterBuilder](#), [IntParameterBuilder](#), [StringParameterBuilder](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

GetParameter()

Allows the user to get the fully built Parameter object to pass to the OptionBuilder class.

```
public virtual Parameter GetParameter()
```

Returns

[Parameter](#)

IsEnum()

Specifies the type of the parameter as enum.

```
public EnumParameterBuilder IsEnum()
```

Returns

[EnumParameterBuilder](#)

IsFloat()

Specifies the type of the parameter as float.

```
public FloatParameterBuilder IsFloat()
```

Returns

[FloatParameterBuilder](#)

IsInt()

Specifies the type of the parameter as int.

```
public IntParameterBuilder IsInt()
```

Returns

[IntParameterBuilder](#)

IsRequired()

Specifies whether the option is required. If not set, the user will be issued a warning.

```
public ParameterBuilder IsRequired()
```

Returns

[ParameterBuilder](#)

IsString()

Specifies the type of the parameter as string.

```
public StringParameterBuilder IsString()
```

Returns

[StringParameterBuilder](#)

WithLambdaValidation()

Allows the user to enter a lambda function for parameter verification.

```
public virtual ParameterBuilder WithLambdaValidation()
```

Returns

[ParameterBuilder](#)

Class StringParameter

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll

Encapsulates string parameters for options.

```
public class StringParameter : Parameter
```

Inheritance

[object](#)  ← [Parameter](#) ← StringParameter

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

Value()

Returns the value of the string parameter.

```
public override string Value()
```

Returns

[string](#) 

Class StringParameterBuilder

Namespace: [ArgumentAPI](#)

Assembly: API Implementation.dll








Provides method overrides for string type parameters.

```
public class StringParameterBuilder : ParameterBuilder
```

Inheritance

[object](#)  ← [ParameterBuilder](#) ← StringParameterBuilder

Inherited Members

[ParameterBuilder.IsRequired\(\)](#), [ParameterBuilder.IsString\(\)](#), [ParameterBuilder.IsEnum\(\)](#), [ParameterBuilder.IsInt\(\)](#), [ParameterBuilder.IsFloat\(\)](#), [ParameterBuilder.WithLambdaValidation\(\)](#), [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

GetParameter()

Returns the built StringParameter object.

```
public override StringParameter GetParameter()
```

Returns

[StringParameter](#)

WithDefaultValue(string)

Sets the default value for the string parameter. A warning is thrown if the value does not meet the criteria defined by the lambda validation function.

```
public StringParameterBuilder WithDefaultValue(string value)
```

Parameters

value [string](#)

Returns

[StringParameterBuilder](#)

WithLambdaValidation(Func<string, bool>?)

Accepts a lambda function for string validation.

```
public StringParameterBuilder WithLambdaValidation(Func<string, bool>? validation = null)
```

Parameters

validation [Func](#) <[string](#), [bool](#)>

Returns

[StringParameterBuilder](#)