

# Namespace Extism

## Classes

### [HttpRequest](#)

An HTTP request

### [HttpResponse](#)

Response from an HTTP call

### [MemoryBlock](#)

A block of allocated memory.

### [Pdk](#)

Provides interop functions for communication between guests and the host.

## Enums

### [HttpMethod](#)

HTTP Method

### [LogLevel](#)

Log level

# Enum HttpMethod

Namespace: [Extism](#)

Assembly: Extism.Pdk.dll

HTTP Method

```
public enum HttpMethod
```

## Fields

DELETE = 3

DELETE

GET = 0

GET

HEAD = 4

HEAD

PATCH = 5

PATCH

POST = 1

POST

PUT = 2

PUT

# Class HttpRequest


Namespace: [Extism](#)

Assembly: Extism.Pdk.dll








An HTTP request

```
public class HttpRequest
```

## Inheritance

[object](#)  ← HttpRequest

## Inherited Members

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

## Constructors

### HttpRequest(string)

```
public HttpRequest(string url)
```

#### Parameters

*url* [string](#) 

### HttpRequest(Uri)

```
public HttpRequest(Uri url)
```

#### Parameters

*url* [Uri](#) 

# Properties

## Body

An optional body.

```
public byte[] Body { get; set; }
```

## Property Value

[byte](#)[]

## Headers

HTTP Headers

```
public Dictionary<string, string> Headers { get; }
```

## Property Value

[Dictionary](#)<[string](#), [string](#)>

## Method

HTTP method

```
public HttpMethod Method { get; set; }
```

## Property Value

[HttpMethod](#)

## Url

HTTP URL

```
public Uri Url { get; set; }
```

Property Value

[Uri](#)

# Class HttpResponse

Namespace: [Extism](#)

Assembly: Extism.Pdk.dll

Response from an HTTP call

```
public class HttpResponse : IDisposable
```








## Inheritance

[object](#)  ← HttpResponse

## Implements

[IDisposable](#) 

## Inherited Members

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

## Constructors

### HttpResponse(MemoryBlock, ushort)

```
public HttpResponse(MemoryBlock memory, ushort status)
```

## Parameters

**memory** [MemoryBlock](#)

**status** [ushort](#) 

## Properties

### Body

Body of the HTTP response

```
public MemoryBlock Body { get; }
```

Property Value

[MemoryBlock](#)

## Status

HTTP Status Code

```
public ushort Status { get; set; }
```

Property Value

[ushort](#)

## Methods

### Dispose()

Frees the current memory block.

```
public void Dispose()
```

# Enum LogLevel

Namespace: [Extism](#)

Assembly: Extism.Pdk.dll

Log level

```
public enum LogLevel
```

## Fields

Debug = 1

Debug

Error = 3

Error

Info = 0

Information

Warn = 2

Warning



# Class MemoryBlock

Namespace: [Extism](#)

Assembly: Extism.Pdk.dll

A block of allocated memory.

```
public class MemoryBlock : IDisposable
```








## Inheritance

[object](#)  ← MemoryBlock

## Implements

[IDisposable](#) 

## Inherited Members

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

## Constructors

### MemoryBlock(ulong, ulong)

```
public MemoryBlock(ulong offset, ulong length)
```

## Parameters

offset [ulong](#) 

length [ulong](#) 

## Properties

### Empty

An Empty memory block.

```
public static MemoryBlock Empty { get; }
```

Property Value

[MemoryBlock](#)

## IsEmpty

Is block empty.

```
public bool IsEmpty { get; }
```

Property Value

[bool](#)

## Length

Length of block in bytes

```
public ulong Length { get; }
```

Property Value

[ulong](#)

## Offset

Starts address of block

```
public ulong Offset { get; }
```

Property Value

[ulong](#)

# Methods

## CopyTo(Span<byte>)

Copies the contents of a memory block into a buffer.

```
public void CopyTo(Span<byte> buffer)
```

### Parameters

buffer [Span](#) [<byte>](#)

### Exceptions

[InvalidOperationException](#)

## Dispose()

Frees the current memory block.

```
public void Dispose()
```

## Find(ulong)

Finds a memory block based on its start address.

```
public static MemoryBlock Find(ulong offset)
```

### Parameters

offset [ulong](#)

### Returns

[MemoryBlock](#)

## ReadBytes()

Reads the current memory block as a byte array.

```
public byte[] ReadBytes()
```

Returns

[byte](#)[]

## ReadString()

Reads the current memory block as a UTF8 encoded string.

```
public string ReadString()
```

Returns

[string](#)

## WriteBytes(ReadOnlySpan<byte>)

Writes a byte buffer to this memory block.

```
public void WriteBytes(ReadOnlySpan<byte> bytes)
```

Parameters

bytes [ReadOnlySpan](#)<[byte](#)>

Exceptions

[IndexOutOfRangeException](#)

## WriteString(string)

Writes a string to the current memory block.

```
public void WriteString(string text)
```

## Parameters

text [string](#) 

# Class Pdk

Namespace: [Extism](#)

Assembly: Extism.Pdk.dll








Provides interop functions for communication between guests and the host.

```
public static class Pdk
```

## Inheritance

[object](#)  ← Pdk

## Inherited Members

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

## Methods

### Allocate(ReadOnlySpan<byte>)

Allocate a byte buffer into memory.

```
public static MemoryBlock Allocate(ReadOnlySpan<byte> buffer)
```

## Parameters

**buffer** [ReadOnlySpan](#)  <[byte](#)  >

The byte buffer to allocate into memory.

## Returns

[MemoryBlock](#)

A [MemoryBlock](#) instance representing the allocated memory.

## Allocate(string)

Encodes a string as UTF-8 and stores it in memory.

```
public static MemoryBlock Allocate(string data)
```

### Parameters

data [string](#)

The string to allocate into memory.

### Returns

[MemoryBlock](#)

A [MemoryBlock](#) instance representing the allocated memory.

## Allocate(ulong)

Allocate a block of memory with the specified length.

```
public static MemoryBlock Allocate(ulong length)
```

### Parameters

length [ulong](#)

The size of the memory block in bytes.

### Returns

[MemoryBlock](#)

A [MemoryBlock](#) instance representing the allocated memory.

## GetInput()

Read the input data sent by the host.

```
public static byte[] GetInput()
```

Returns

[byte](#)[]

The input data as a byte array.

## GetInputString()

Read the input data sent by the host as a UTF-8 encoded string.

```
public static string GetInputString()
```

Returns

[string](#)

## Log(LogLevel, MemoryBlock)

Log a message with the specified log level to the host.

```
public static void Log(LogLevel level, MemoryBlock block)
```

Parameters

**level** [LogLevel](#)

The log level for the message.

**block** [MemoryBlock](#)

The memory block containing the log message.

## Log(LogLevel, string)

Logs a message to the host.



```
public static void Log(LogLevel level, string message)
```

## Parameters

level [LogLevel](#)

message [string](#)

## RemoveVar(string)

Remove a variable from host memory.

```
public static void RemoveVar(string key)
```

## Parameters

key [string](#)

The key of the variable to remove.

## SendRequest(HttpRequest)

Send an HTTP request synchronously and get the response from the host.

```
public static HttpResponseMessage SendRequest(HttpRequest request)
```

## Parameters

request [HttpRequest](#)

The HTTP request to send.

## Returns

[HttpResponse](#)

The HTTP response received from the host.

## SetError(string)

Set plugin error

```
public static void SetError(string errorMessage)
```

Parameters

errorMessage [string](#)

## SetOutput(MemoryBlock)

Set the output data to be sent back to the host.

```
public static void SetOutput(MemoryBlock block)
```

Parameters

block [MemoryBlock](#)

The memory block containing the output data.

## SetOutput(ReadOnlySpan<byte>)

Set the output data to be sent back to the host as a byte buffer.

```
public static void SetOutput(ReadOnlySpan<byte> data)
```

Parameters

data [ReadOnlySpan](#) <[byte](#)>

The byte buffer to set as output data.

## SetOutput(string)

Set the output data to be sent back to the host as a UTF-8 encoded string.

```
public static void SetOutput(string data)
```

## Parameters

data [string](#)

The UTF-8 encoded string to set as output data.

## SetVar(string, MemoryBlock)

Set a var that will be persisted by the host.

```
public static void SetVar(string key, MemoryBlock value)
```

## Parameters

key [string](#)

value [MemoryBlock](#)

## SetVar(string, ReadOnlySpan<byte>)

Set a variable value persisted by the host.

```
public static void SetVar(string key, ReadOnlySpan<byte> bytes)
```

## Parameters

key [string](#)

The key for the persisted variable.

bytes [ReadOnlySpan](#) <[byte](#)>

The byte buffer to set as the variable value.

## SetVar(string, string)

Set a variable value persisted by the host.

```
public static void SetVar(string key, string value)
```

## Parameters

key [string](#)

The key for the persisted variable.

value [string](#)

A string value that will be UTF8 encoded.

## TryGetConfig(string, out string)

Try to get a configuration value from the host.

```
public static bool TryGetConfig(string key, out string value)
```

## Parameters

key [string](#)

The key for the configuration value.

value [string](#)

The retrieved configuration value as a string.

## Returns

[bool](#)

True if the configuration value was retrieved successfully; otherwise, false.

## TryGetVar(string, out MemoryBlock)

Log a message with the specified log level to the host.

```
public static bool TryGetVar(string key, out MemoryBlock block)
```

## Parameters

key [string](#) 

The log level for the message.

block [MemoryBlock](#)

The log message as a string.

## Returns

[bool](#) 