Iliad developer's guide

Version 0.9.1
23 octobre 2010

Nicolas Petton and Steven Costiou

Contents

Introduction

Iliad is a Smalltalk web application framework. Heavily based on reusable stateful widgets, Iliad lets you build powerful dynamic applications easily. It allows you to trigger Smalltalk code when an user clicks on a link or submits a form instead of bothering you with low level details.

Thanks to its JavaScript layer Iliad automatically uses AJAX requests to update the client state, and it will nicely degrade to normal requests if JavaScript is not enabled, so you don't have to bother about that either.

Resources

Here is a short list of Iliad related resources:

- the main website of Iliad (http://www.iliadproject.org);
- Iliad mailing-list (http://groups.google.fr/group/iliad);
- the issue tracker (http://code.google.com/p/iliadproject/issues).

A work in progress

Iliad's documentation is evolving. We are looking for contributors to help us improve this documentation.

Chapter 1

Downloading and installing Iliad

Iliad is currently available for GNU Smalltalk, Squeak and Pharo. In this chapter you will find detailed installation instructions for each supported Smalltalk dialect.

1.1 Installing on GNU Smalltalk

gst-package

With *GNU Smalltalk* 3.2 or newer, `gst-package` can be used to download and install Iliad.

```
~$ gst-package --download grease -t ~/.st
~$ gst-package --download iliad -t ~/.st
```

Git

You can download Iliad by cloning the *Git* repository.

```
~$ git clone http://github.com/NicolasPetton/Grease.git
~$ cd Grease
~$ gst-package -t ~/.st ./package.xml

~$ git clone git://github.com/NicolasPetton/iliad-stable.git
~$ cd iliad-stable
~$ ./scripts/make_packages
```

1.2 Installing on Pharo Smalltalk

Using Metacello

Download `ConfigurationOfIliad` package from the Metacello repository from squeaksource :

```
MCHttpRepository
location: 'http://www.squeaksource.com/MetacelloRepository'
user: ''
password: ''
```

Open a workspace and *do-it*:

```
ConfigurationOfIliad load
```

Manually

Load Swazoo 2.2 (http://www.squeaksource.com/Swazoo) and Grease (http://www.squeaksource.com/Seaside30) from SqueakSource. Load packages from the SqueakSource repository (http://www.squeaksource.com/iliad) in the following order:

- Iliad-Core
- Iliad-Swazoo
- Iliad-More-UI
- Iliad-More-Rss
- Iliad-More-Formula
- Iliad-More-Comet
- Iliad-More-Examples
- Iliad-More-Tests

1.3 Loading and starting Iliad

Iliad is webserver agnostic. However, only one server adapter for Swazoo (http://www.swazoo.org) has been written so far. This adapter is available for Pharo Smalltalk and GNU Smalltalk.

1.3.1 GNU Smalltalk

The start script

The git repository of Iliad includes a `start` script, which will create an image with all required packages loaded and start Iliad with gst-remote.

```
~$ ./scripts/start -p PORT
```

gst's REPL

You can load Iliad packages into a GNU Smalltalk image with the PackageLoader. When using gst's REPL, you will have to suspend the active process to make Swazoo listen to a given port:

```
st> PackageLoader fileInPackage: 'Iliad'
st> Iliad.SwazooIliad startOn: 8888
st> processor activeProcess suspend
```

With gst-remote

You can create a bash script and use it to load the desired packages and start Iliad:

```
#!/bin/bash

# load packages into a new image named iliad.im
echo "PackageLoader fileInPackage: 'Iliad'. ObjectMemory snapshot: 'iliad.im'" | gst

# start gst-remote with the image
gst-remote --server -I iliad.im &
sleep 2

# start Swazoo on port 8888
gst-remote --eval "Iliad.SwazooIliad startOn: 8888"
```

1.3.2 Pharo Smalltalk

To start Swazoo with the Iliad adapter, *do-it*:

```
SwazooIliad startOn: 8888
```

1.4 Testing the installation

Open 'http://localhost:8888/browse' ('http://localhost:8888/browse') in your browser. You should see the list of loaded Iliad applications.

Chapter 2

Short tutorial

In this short tutorial we will build our first Iliad widget: a simple counter. Widgets (see chapter 5 [Widgets] on page 35) are high-level reusable, stateful, graphical objects. We will create the class and a method to build HTML, then add some behavior to increase and decrease the counter.

Our first widget

Widgets are subclasses of ILWidget. Our widget will have a `count` *instance variable*, initiliazed to zero.

```
Iliad.ILWidget subclass: CounterWidget [
    | count |

    initialize [
        super initiliaze.
        count := 0
    ]
]
```

To build HTML Iliad calls the `#contents` method of a widget. This method answers a buildable object: almost always a *block closure* or another widget.

```
contents [
    ^[:e | e h1: count printString]
]
```

Action methods

Now we can add action methods to increase and decrease the counter.

```
increase [
    count := count + 1
]

decrease [
    count := count - 1
]
```

To allow the user to increase or decrease the counter, we build anchors which will call the action methods (see section 3.3 [Actions] on page 27), and modify the `contents` method as follow.

```
contents [
    ^[:e |
        e h1: self count printString.
        e a
            action: [self increase];
            text: ’++’.
        e a
            action: [self decrease];
            text: ’++’]
]
```

To tell Iliad that the state of the counter has changed and that it should be rebuilt, we call its `#markDirty` method in the action methods.

```
increase [
    count := count + 1.
    self markDirty
]

decrease [
    count := count - 1.
    self markDirty
]
```

Using the counter widget in an application

To see our widget in action, we build it in an application. Applications (see chapter 4 [Applications] on page 31) are similar to widgets except that they dispatch requests in *controller methods*, similar to the `#contents` method of widgets. The default controller method is `#index`.

```
Iliad.ILApplication subclass: CounterApplication [

    CounterApplication class >> path [
        ^’counter’
    ]

    counterWidget [
        ^counterWidget ifNil: [counterWidget := CounterWidget new]
    ]

    index [
        <category: 'controllers'>
        ^self counterWidget
    ]
]
```

The class side #path method answers the base path of our application.

Chapter 3

Elements

Elements are low-level stateless objects composed in a tree to build valid (X)HTML. For each element class there is a corresponding HTML tag. This means you will never have to worry about writing valid HTML yourself, but instead use the element API to build HTML in Smalltalk.

`#contents` methods of widgets (see chapter 5 [Widgets] on page 35) or *controller methods* of applications (see chapter 4 [Applications] on page 31) are called by Iliad to build HTML using elements.

3.1 Composing the element tree

Adding elements

`ILHTMLBuilderELement` is the root class of the HTML element hierarchy. It has convenience methods for adding other HTML elements, and manipulate HTML attributes. These convenience methods follow HTML tag and attribute names. You can browse them in the `adding-convenience` protocol. All HTML elements inherit these methods.

The API provides the `#add:` method for adding a child to an element. Convenience methods use this method to compose the tree of elements.

```
p [
    <category: 'adding-conveniance'>
    ^self add: ILParagraphElement new
]
```

The HTML API

Each element in the tree represents a given HTML tag. Each attribute of an element has a corresponding HTML tag attribute. We compose HTML by combining the two kind of methods. As methods are named depending on the HTML tag they represent, building HTML with Iliad looks very similar to actual HTML code.

The following example

```
e p: 'Hello world'.
e br.
e a
    href: 'http://www.iliadproject.org';
    text: 'Iliad website'.
```

Will produce the HTML:

```
<p>Hello world</p>
<br/>
<a href='http://www.iliadproject.org'>Iliad website</a>
```

Most of the time you won't have to learn the HTML API if you already know the HTML tag names or attributes you want to write. If you can't find the name of a convenience method, you can browse the `ILHTMLBuilderELement` class and its subclasses in `Iliad-Core-HTMLElements`.

3.2 Elements and buildables

Buildable objects are high-level stateful graphical objects. They use elements to build themselves as HTML. They must implement the `#buildOn:` method, taking an element as parameter.

A buildable object can be built on an element with the `ILElement>>build:` method.

Default buildable objects in Iliad includes *Block closures* and *Widgets*.

```
div build: [:e | e text: 'hello world'].
div build: myWidget.
```

Buildables should never be built using their `#contents` method or the `ILElement>>add:` method. Therefore, the following example is not valid.

```
e add: myWidget contents. "Not valid"
e build: myWidget. "Valid"
```

Block closures can be used to make building methods simpler by nesting elements.

```
| div a |
div := e div.
a := div a.
a href: 'http://www.iliadproject.org'.
```

```
a img
    src: 'iliad.png';
    alt: 'Iliad logo'.
div h1: 'Iliad rocks!'
```

Can be written:

```
e div build: [:div |
    div a build: [:a |
        a href: 'http://www.iliadproject.org'.
        a img
            src: 'iliad.png';
            alt: 'Iliad logo'].
    div h1: 'Iliad rocks!']
```

And will build the following HTML code:

```
<div>
    <a href='http://www.iliadproject.org'>
        <img src='iliad.png' alt='Iliad logo'/>
    </a>
    <h1>Iliad rocks!</h1>
</div>
```

3.3 Actions

Buildables can do more than just build HTML. They can behave, change their state and react to user interactions through Smalltalk code evaluated in actions.

Instead of having to parse URLs and requests, Iliad uses action objects to trigger Smalltalk code when the user interacts with the page, clicks on a link or submits a form. An *action* is an Iliad object with a Smalltalk block associated to an id, but we will oftenly refer to the block of an action as the action itself.

Actions in anchors

The `ILAnchorElement` class provides the `#action:` method to associate an action block to the anchor.

```
e a
    text: 'Click!';
    action: [Transcript show: 'Anchor clicked.']
```

Because `#action:` will add an `href` attribute with a generated URL, anchors can't have both an associated action and an href.

```
e a
    text: 'Click!';
    action: [Transcript show: 'Anchor clicked.'; cr];
    href: 'http://www/smalltalk.org' "This will override the href set by #action:"
```

The counter widget example in `Iliad-More-Examples` is a good example of actions in anchors.

```
contents [
    ^[:e |
        e h1: self count printString.
        e a
            action: [self increase];
            text: '++'.
        e space.
        e a
            action: [self decrease];
            text: '--']
]
```

Actions in forms

Form elements can also have actions, evaluated when the form is submitted. We can modify the `#contents` method of the counter example to use a form and buttons instead of anchors.

```
contents [
    ^[:e |
        e h1: self count printString.
        e form build: [:form |
            form button
                text: '++';
                action: [self increase].
            form button
                text: '--';
                action: [self decrease]]]
]
```

Some form elements like inputs or textareas take a one argument action block. The argument is the value entered by the user.

```
e form build: [:form |
    form input action: [:val | self doSomethingWith: val].
    form button text: 'Ok']
```

3.4 DOM events

Iliad provides a simple API to associate JavaScript code to DOM events (click, focus, etc.) in the `accessing attributes-events` method protocol. The main method is `#on:add:` which takes two strings as arguments, the first one for the event, and the last one for the associated JavaScript code. Many shortcut methods exist for specific events, like `#onClick:` or `#onFocus:`.

```
e p
    text: 'Click';
    onClick: 'alert("clicked!");'.
```

If an element cannot be associated with a DOM event, an `ILAttributeNotUnderstood` error will be raised.

Evaluating actions on DOM Events

It is also possible to associate actions to DOM events. Names of methods follow the same convention, but end with *do:* and take a block as parameter.

```
e p
    text: 'An action block is evaluated on a click event';
    onClickDo: [Transcript show: 'Clicked!'; cr]
```

Chapter 4

Applications

An application is the root of a buildable object tree composing the user interface. Applications are used to dispatch requests to other buildables, typically widgets (see chapter 5 [Widgets] on page 35), and hold their state. Unlike widgets they don't have one building method but several *controller methods* used to dispatch requests to the right buildable object.

Application instances are handled by the framework. Iliad stores one instance of an application class per session, which will persist through requests.

4.1 The base path

Each application class should have an unique *base path*, a string corresponding to the path of the application. The base path is answered by class method `path`.

```
Iliad.ILApplication subclass: MyApplication [
    MyApplication class >> path [
        "Answer the base path of the application"
        ^'my_application'
    ]
]
```

During a request processing, Iliad will determine to which application the request will be dispatched depending on its base path. If Iliad is running on port 8888, instances of MyApplication will be reached at:

```
http://localhost:8888/my_application
```

By default, `ILApplication`'s path is the empty string. This means the class method `path` must be overriden, else the application will not be reachable.

The base path of an application is an absolute path and can freely be composed with `'/'`. We can for example create two application classes with the same first path fragment: `'/foo/bar'` and `'/foo/baz'`.

4.2 Controller methods

Controller methods are building methods used to dispatch requests. The application will dispatch the request to the method named after the next fragment of the URL. The default controller method is #index. Concrete applications should always override this method to answer a buildable object.

For example, if the base path of the application is 'foo', the application will try to dispatch a request url '/foo/bar' to the controller method #bar. If there is no method matching this rule, #index will be called.

By default, controller methods must be in the 'controllers' method protocol to be allowed to be called as controller methods (see section 4.3 [Filtering controllers] on the facing page).

A 'hello world' application

At this point we can write a 'hello world' application, with a base path and an #index controller.

```
Iliad.ILApplication subclass: HelloWorldApplication [

    HelloWorldApplication class >> path [
        <category: 'accessing'>
        ^'hello_world'
    ]

    index [
        <category: 'controllers'>
        ^[:e | e h1: 'hello world']
    ]
]
```

After starting Iliad on port 8888, the application can be browsed at http://localhost:8888/hello_world (http://localhost:8888/hello_world). As expected, the application will dispatch both /hello_world and /hello_world/index to the #index controller.

A multi-counter application

Let's improve the counter application written in the tutorial (see chapter 2 [Short tutorial] on page 21) and make it a multi-counter application. The application will build several counter widgets in the #index controller.

```
Iliad.ILApplication subclass: CounterApplication [
    CounterApplication class >> path [
        <category: 'accessing'>
        ^'counter'
    ]
    initialize [
        <category: 'initialization'>
        super initialize.
        counters := Array
            with: CounterWidget new
            with: CounterWidget new
            with: CounterWidget new
    ]

    index [
        <category: 'controllers'>
        ^[:e |
            counters do: [:each | e build: each]]
    ]
]
```

To persist through requests, widgets must be stored in instance variables. In this example counters are stored at initialization in a collection.

4.3 Filtering controllers

The use of a method of an application as a controller method is restricted by the *selector filter*. Each application class has its own selector filter. Each time an application dispatches a request to a controller method, the selector filter block is evaluated against a selector to determine if the corresponding method is allowed to be used as controller or not. The default selector filter allows all methods in the `'controllers'` protocol, but you may want to filter selectors differently.

The default selector filter can be overriden in the class side `#defaultSelectoreFilter` method, or you can plug in a new selector filter with the class side `#selectorFilter:`. A selector filter block must be a one argument block and answer a boolean.

In the following example, the only selector allowed is `#index`, all other methods will be forbidden.

```
MyApplication selectorFilter: [:selector | selector == #index]
```

4.4 Custom request dispatching

`ILApplication` provides the `#dispatchOverride` hook method to handle special cases for dispatching a request. Subclasses of `ILApplication` should call `super` first to see if it was already handled.

The behavior of #dispatchOverride is very simple. Before dispatching a request to a controller method, #dispatchOverride is called. If the answer is anything but nil, the application will consider the request as handled and will not dispatch it to a controller method.

See section 7.2 [Using a custom session class] on page 40, for a simple usage example of this method.

4.5 Updating the page

During the generation of a page for a request processing, the application can update any part of it using the #updatePage: method. This method will be called for each new request excepting AJAX ones. In this method we can for example change the title of the page, update its <head> section, etc.

#updatePage: should not be used to update contents of widgets included in the <body> section, as widgets may not be built yet when Iliad calls #updatePage:.

The argument sent to #updatePage: is an instance of Iliad.ILPage. This class has accessors for both the <head> section and the <body> section of the page, respectively instances of Iliad.ILHeadElement and Iliad.ILBodyElement.

```
updatePage: aPage [
    <cateogry: 'updating'>
    super updatePage: aPage.
    aPage head title: 'The multi-counter example'.
    aPage head stylesheet href: '/path/to/my/style.css'
]
```

super should always be called, as Iliad uses Javascript files (see chapter 8 [The Javascript layer] on page 43) by default to send AJAX requests automatically. See chapter 9 [Serving static files] on page 45, for more details about how to serve static files with Iliad.

Chapter 5

Widgets

Iliad web pages are composed of embedded Widgets. Widgets are high-level, reusable graphical objects. Widgets are not only about building HTML, they also hold the state of web applications and will persist through requests. `ILWidget` is probably the most important class of Iliad, as widgets will compose most of your user interfaces and control flow.

5.1 Maintaining the user interface state

To store the user interface state, applications and widgets must know and store their children. Applications themselves are stored in sessions (see chapter 7 [Managing sessions] on page 39). Widgets are stored in instance variables in a tree of widgets by their parent, applications or other widgets. During a request processing, after the actions evaluation step, widgets are rebuilt and sent to the client in HTML or JSON (see section 5.2 [Maintaining the client state] on this page).

5.2 Maintaining the client state

When updating the client state, refreshing the entire page to update some widgets - therefore rebuilding the entire application with all its children widgets - would be a waste of bandwidth and time. Iliad is designed to automatically build full-blown AJAX applications without hard-coding Javascript inside the HTML page. Because Iliad uses a thin Javascript layer, your content will automatically degrade on clients that don't have Javascript enabled.

With Javascript enabled web browsers, Iliad automatically uses AJAX to update the client state. The javascript layer (see chapter 8 [The Javascript layer] on page 43) catches clicks on anchors with actions and form submissions, and send to the server a XmlHTTPRequest. During the request processing, widgets which state has changed are sent back to the browser in JSON. The

Javascript layer will then update the page with the new data to keep the client state up-to-date.

The `#markDirty` method is used to inform Iliad that a widget's state has changed. It will rebuild it and send it's new HTML contents in JSON to the client.

5.3 Dependent widgets

5.4 Interacting with other widgets

5.5 Sequencing with `ILSequence`

5.6 Widget decorators

5.7 Redirecting

5.8 Updating the `<head>` section of the page

Chapter 6

Routes and urls

TODO

- The ILRoute object
- Routers
- ILUrlBuilder and the base url

Chapter 7

Managing sessions

Session objects are automatically instanciated by the framework during the first interaction with an Iliad application. Sessions hold the state of the web application: application instances, actions and states of widgets. They can also be used to keep globally accessible and shared informations among the user interface. A session persists as long as it is active, i.e., as long as the user interacts with the application. After a timeout, if the user is not interacting with the application anymore, the session will expire.

7.1 The session manager

The *session manager* is the object responsible for managing sessions throughout every Iliad application. The current session manager can be accessed through the class side method `ILSessionManager>>current`. `ILSessionManager` implements methods for internal session management. These methods are part of the Iliad core, and are intended to be used while developping, deploying or debugging an Iliad application. Manually removing or invalidate data from sessions is not recommended.

Convenience methods

The session manager can at will remove all sessions it contains:

```
Iliad.ILSessionManager current removeAllSessions
```

Expired sessions are automatically removed from the session manager from time to time. However, you can explicitely remove all expired sessions with:

```
Iliad.ILSessionManager current removeExpiredSessions
```

Accessing sessions

Sessions can be accessed with the `#sessions` accessor. Iteration through sessions can be done with `#sessionsDo:`. Note that widgets and applications should never try to access the current session through the session manager, but instead use their `#session` accessor.

Accessing the current session from a debugger or an inspector is not always possible since it is only accessible in the context of a request processing. A workaround is to inspect the session object while processing the request by adding the following code inside the `#contents` method of a widget or inside a controller method of an application:

```
self session inspect
```

7.2 Using a custom session class

`ILSession` is the default session class and can be subclassed to store special informations or handle actions. As an example we will write a session class with very basic login facilities and use it in a simple application.

```
Iliad.ILSession subclass: MySession [
    | username |

    username [
        ^username
    ]

    username: aString [
        username := aString
    ]
]
```

We can create an application class that will only be accessible to logged in users:

```
Iliad.ILApplication subclass: MyApplication [

    MyApplication class >> path [^'my_app']
]
```

A login form with validation, registration, password recovery, etc. should be built using a custom widget. In a matter of simplicity, here we will just write the form in a building method in the application.

```
loginContents [
    <category: 'building'>
    ^[:e |
        e form build: [:form |
            form input action: [:val | self login: val].
            form button text: 'Login']]
]

login: aString [
    <category: 'actions'>
    self session username: aString.
    self redirectToCurrentController
]
```

Testing if the user is logged in or not can happen in a controller method. However, if we want the application to globally accessible to logged in users only, we can use the hook method `#dispatchOverride` (see section 4.4 [Custom request dispatching] on page 33).

```
dispatchOverride [
    <category: 'dispatching'>
    ^self session username
        ifNil: [self loginContents]
        ifNotNil: [super dispatchOverride]
]
```

7.3 Handling session expiration

Sessions expire after some time if the user is not interacting with the application. The default timeout is set to 1800 seconds, and can be changed with: `ILSession>>expirySeconds:`.

When a session expires, Iliad calls its `#onExpire` method. If you use your own session class, you can override this method to trigger events for session expiration: close database connections, displaying session expiration informations to the user, etc. To illustrate this behavior, we will reuse our session class and logout the user when its session expires.

```
onExpire [
    <category: 'events'>
    self username: nil.
    super onExpire
]
```

Chapter 8

The Javascript layer

Chapter 9

Serving static files

Chapter 10

RSS

Chapter 11

Formula

Chapter 12

Deploying Iliad applications

TODO

- Configuring apache
- Serving static files
- ILUrlBuilder and rewrite rules

Chapter 13

Iliad class library reference

13.1 Iliad.IdentityBag

Defined in namespace Iliad
Superclass: Bag
Category:

13.2 Iliad.ILAction

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions

13.2.1 Iliad.ILAction: accessing

block Answer ‘block’.

block: aBlock
Not commented.

key Answer ‘key’.

key: anId
Not commented.

value Answer ‘value’.

value: anObject
Not commented.

13.2.2 Iliad.ILAction: converting

respondOn: aResponse
Not commented.

13.2.3 Iliad.ILAction: evaluating

evaluate
Not commented.

13.3 Iliad.ILActionRegistry

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions
I implement a registry of actions.

13.3.1 Iliad.ILActionRegistry: accessing

actionAt: aKey
Not commented.

actions Not commented.

13.3.2 Iliad.ILActionRegistry: actions

evaluateActionKey: aString
Not commented.

register: anAction
Not commented.

unregister: anAction
Not commented.

unregisterAllActions
Not commented.

13.4 Iliad.ILAnchorElement

Defined in namespace Iliad
Superclass: Iliad.ILLinkableElement
Category: Iliad-Core-XHTMLElements

13.4.1 Iliad.ILAnchorElement: accessing

action: aBlock
Not commented.

action: aBlock hash: aString
Not commented.

tag Answer ''a".

13.4.2 Iliad.ILAnchorElement: accessing attributes

tabindex: anInteger
Not commented.

13.4.3 Iliad.ILAnchorElement: accessing attributes-imagemap

circleShape
Not commented.

coords: aString
Not commented.

defaultShape
Not commented.

polyShape
Not commented.

rectShape
Not commented.

shape: aString
Not commented.

13.5 Iliad.ILAnswerHandler

Defined in namespace Iliad
Superclass: Iliad.ILDecorator
Category: Iliad-Core-Buildables

I am a special decorator for Widgets, used to handle widget answers. See #handleAnswer: and ILWidget>>show:onAnswer:

13.5.1 Iliad.ILAnswerHandler: accessing

action Answer 'action'.

action: anAction
Not commented.

13.5.2 Iliad.ILAnswerHandler: decorations

handleAnswer: anAnswer
Not commented.

13.6 Iliad.ILAppendDelegator

Defined in namespace Iliad
Superclass: Iliad.ILDelegator
Category: Iliad-Core-Buildables

13.6.1 Iliad.ILAppendDelegator: building

contents
Not commented.

13.7 Iliad.ILApplication

Defined in namespace Iliad
Superclass: Iliad.ILBuildable
Category: Iliad-Core-Buildables

I am the Iliad implementation of an application.

I am is the root object of a buildable object tree. Applications have a set of controllers, methods used to dispatch requests to the corresponding sub-tree of buildable objects (oftenly a composition of stateful widgets).

In concrete subclasses, the class method #path should return the base path (string) for the application.

""""""""""""""""""""""""""" " Applications & UI state " """""""""""""""""""""""""""

You don't have to bother about instantiating applications, the framework will handle session and application instances. Application instances are stored in sessions. Each session stores one instance of the same application class.

Root widgets should be stored in applications to keep their state across requests.

"""""""""""""""""""""" " Controller methods " """"""""""""""""""""""

Like widgets, I am stateful. Unlike widgets I know how to dispatch a request with #dispatch : the controller method corresponding to the url will be called.

Controller methods must: - answer a buildable object (a block closure or an instance of ILWidget for example). - be in the 'controllers' method protocol (with the default selector filter)

The default controller method is #index.

"""""""""""""""""" " selectorFilter " """"""""""""""""""

The class inst var <selectorFilter> is used to filter controller methods. By default it allows all methods in the 'controllers' protocol.

Alternatively, you can override the class method #defaultSelectorFilter to supply your own selectorFilter or plug it in using the class method #selectorFilter:

13.7.1 Iliad.ILApplication class: accessing

absolutePath

Not commented.

path Base path of the application. Override this method in concrete subclasses. It should return a string

selectorFilter
Not commented.

selectorFilter: aBlock
Not commented.

13.7.2 Iliad.ILApplication class: defaults

defaultSelectorFilter
Override this method to supply your own selectorFilter or plug it in using #selectorFilter:

13.7.3 Iliad.ILApplication: accessing

model Answer ‘model’.

model: anObject
Not commented.

page Answer ‘page’.

selectorFilter
Not commented.

widgetFor: aBuildable
Convenience method. This is useful for building anonymous widgets. ex: myWidget := self widgetFor: [:e | e h1: ’Hello world!’]

13.7.4 Iliad.ILApplication: building

buildContents
Call #dispatch. A buildable is expected from #dispatch

13.7.5 Iliad.ILApplication: controllers

index default view method

13.7.6 Iliad.ILApplication: converting

respondOn: aResponse
Not commented.

13.7.7 Iliad.ILApplication: defaults

defaultPageClass
Answer 'ILHTMLPage'.

rootElementClass
Answer 'ILHTMLBuilderElement'.

13.7.8 Iliad.ILApplication: dispatching

allowedSelector: aSelector
Answer true if <aSelector> is ok to call from a URL. Default implementation is to use the pluggable filter block.

dispatch
Dispatch to correct controller method. If dispatchOverride returns something different from nil, consider it handled.

dispatchOn: aMethod
Dispatch to correct method: - If <aMethod> is empty we call #index - If the selector is allowed to be executed then we just call it

dispatchOverride
Handle special urls. Subclass implementors should call super first and see if it was handled.

13.7.9 Iliad.ILApplication: redirecting

respond: aBlock
Abort all other request handling

returnResponse: aResponse
Abort all other request handling

13.7.10 Iliad.ILApplication: updating

updateBaseUrl: anUrl
Answer the receiver.

updateFromRoute: aRoute
Answer the receiver.

updatePage: aPage
Override to add elements to aPage. super should always be called

13.8 Iliad.ILApplicationHandler

Defined in namespace Iliad
Superclass: Iliad.ILRequestHandler
Category: Iliad-Core-RequestHandlers

13.8.1 Iliad.ILApplicationHandler: initialization

initialize
Not commented.

13.8.2 Iliad.ILApplicationHandler: request handling

evaluateActions
Not commented.

handleRequest
Not commented.

13.8.3 Iliad.ILApplicationHandler: responding

produceResponse
Not commented.

13.8.4 Iliad.ILApplicationHandler: testing

isRequestValid
Not commented.

shouldRedirect
Not commented.

shouldRespondInJson
Not commented.

shouldReturnEmptyResponse
Not commented.

13.9 Iliad.ILAreaElement

Defined in namespace Iliad
Superclass: Iliad.ILClosingElement
Category: Iliad-Core-XHTMLElements

13.9.1 Iliad.ILAreaElement: accessing

tag Answer ''area".

13.9.2 Iliad.ILAreaElement: accessing attributes

accesskey: aCharacter
Not commented.

alt: aString
Not commented.

href: aString
Not commented.

nohref Not commented.

tabindex: anInteger
Not commented.

13.9.3 Iliad.ILAreaElement: accessing attributes-imagemap

circleShape
Not commented.

coords: aString
Not commented.

defaultShape
Not commented.

polyShape
Not commented.

rectShape
Not commented.

shape: aString
Not commented.

13.10 Iliad.ILAttributeError

Defined in namespace Iliad
Superclass: Grease.Grease.GRError
Category: Iliad-Core-Elements

13.11 Iliad.ILAttributeNotUnderstood

Defined in namespace Iliad
Superclass: Grease.Grease.GRError
Category: Iliad-Core-Elements

13.11.1 Iliad.ILAttributeNotUnderstood class: instance creation

element: anElement attribute: aString
Not commented.

13.11.2 Iliad.ILAttributeNotUnderstood: accessing

attribute
Answer ‘attribute’.

attribute: anObject
Not commented.

element Answer ‘element’.

element: anElement
Not commented.

13.11.3 Iliad.ILAttributeNotUnderstood: exceptionDescription

defaultAction
Answer the receiver.

isResumable
Answer ‘true’.

13.11.4 Iliad.ILAttributeNotUnderstood: printing

messageText
Not commented.

13.12 Iliad.ILBodyElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.12.1 Iliad.ILBodyElement: accessing

tag Answer ‘’body”.

13.12.2 Iliad.ILBodyElement: addessing attributes-events

onLoad: aString
Not commented.

onUnload: aString
Not commented.

13.13 Iliad.ILBreakElement

Defined in namespace Iliad
Superclass: Iliad.ILClosingElement
Category: Iliad-Core-XHTMLElements

13.13.1 Iliad.ILBreakElement: accessing

tag Answer ''br''.

13.13.2 Iliad.ILBreakElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.14 Iliad.ILBuildable

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Buildables

I am an abstract buildable object. My subclasses must override #build method, which should return an instance of a subclass of Iliad.ILElement.

13.14.1 Iliad.ILBuildable: accessing

children
Not commented.

router Not commented.

13.14.2 Iliad.ILBuildable: accessing attributes

attributeAt: aSymbol
Not commented.

attributeAt: aSymbol ifAbsentPut: aBlock
Not commented.

attributeAt: aSymbol put: anObject
Not commented.

attributes
Not commented.

13.14.3 Iliad.ILBuildable: actions

send: aSymbol
Not commented.

send: aSymbol to: anObject
Not commented.

send: aSymbol to: anObject arguments: anArray
Not commented.

13.14.4 Iliad.ILBuildable: building

build Not commented.

buildContents
Override this method in subclasses. It must answer an Element

buildOn: anElement
Not commented.

registerChild: aBuildable
Not commented.

13.14.5 Iliad.ILBuildable: converting

respondOn: aResponse
Not commented.

13.14.6 Iliad.ILBuildable: printing

printHtmlString
Not commented.

13.14.7 Iliad.ILBuildable: redirecting

redirectTo: anUrlString
Abort all other request handling. Redirect to anUrlString

redirectToApplication: aClass
Abort all other request handling. Redirect to the index method of <aClass>

redirectToApplication: aClass controller: aString
Abort all other request handling. Redirect to the controller named <aString> of <aClass>

redirectToCurrentController
Abort all other request handling. Redirect to the current controller method

redirectToIndex
Abort all other request handling. Redirect to the index method of this class

redirectToLocal: aString
Abort all other request handling. Make a redirection to another controller method in this application

13.15 Iliad.ILButtonElement

Defined in namespace Iliad
Superclass: Iliad.ILFormElementElement
Category: Iliad-Core-XHTMLElements

13.15.1 Iliad.ILButtonElement: accessing

tag Answer ‘’button”.

13.15.2 Iliad.ILButtonElement: accessing attributes

beButton
Not commented.

beReset
Not commented.

beSubmit
Not commented.

type: aString
Not commented.

value: aString
Not commented.

13.15.3 Iliad.ILButtonElement: printing

beforePrintHtml
Not commented.

13.16 Iliad.ILCheckboxElement

Defined in namespace Iliad
Superclass: Iliad.ILFormElementElement
Category: Iliad-Core-XHTMLElements

13.16.1 Iliad.ILCheckboxElement: accessing

tag Answer ‘’input”.

13.16.2 Iliad.ILCheckboxElement: accessing attributes

action: aBlock
Not commented.

beChecked
Not commented.

checked: aBoolean
Not commented.

13.16.3 Iliad.ILCheckboxElement: printing

addHiddenInput
Not commented.

beforePrintHtml
Not commented.

13.17 Iliad.ILClosingElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.17.1 Iliad.ILClosingElement: printing

printHtmlOn: aStream
Not commented.

13.18 Iliad.ILComposite

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Utilities

13.18.1 Iliad.ILComposite: accessing

children
Not commented.

next Answer ‘next’.

next: aComposite
Not commented.

13.18.2 Iliad.ILComposite: adding

add: aComposite
Not commented.

13.18.3 Iliad.ILComposite: comparing

= anObject
Not commented.

hash Not commented.

13.18.4 Iliad.ILComposite: enumerating

allChildrenDo: aBlock
Not commented.

childrenDo: aBlock
Not commented.

13.19 Iliad.ILConditionalCommentElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.19.1 Iliad.ILConditionalCommentElement: accessing

conditions
Not commented.

13.19.2 Iliad.ILConditionalCommentElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.19.3 Iliad.ILConditionalCommentElement: conditions

gt Greater than

ie Not commented.

ie5 Not commented.

ie50 Not commented.

ie55 Not commented.

ie6 Not commented.

ie7 Not commented.

ie8 Not commented.

lt Less than

not Not

13.19.4 Iliad.ILConditionalCommentElement: printing

printCloseTagOn: aStream
Not commented.

printOpenTagOn: aStream
Not commented.

13.20 Iliad.ILConfirmationWidget

Defined in namespace Iliad
Superclass: Iliad.ILWidget
Category: Iliad-Core-Buildables

13.20.1 Iliad.ILConfirmationWidget: accessing

confirmationString
Answer ‘confirmationString’.

confirmationString: aString
Not commented.

13.20.2 Iliad.ILConfirmationWidget: building

contents
Not commented.

13.21 Iliad.ILContext

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions
I store current request context objects:

- the router - the session - the request itself

I can be accessed through ILObject>>context

13.21.1 Iliad.ILContext: accessing

application
Not commented.

attributes
Not commented.

baseUrl Not commented.

builtWidgets
Not commented.

previousStateRegistry
Not commented.

request Answer ‘request’.

request: aRequest
Not commented.

router Not commented.

session Answer ‘session’.

session: aSession
Not commented.

stateRegistry
Not commented.

urlBuilder
Not commented.

13.21.2 Iliad.ILContext: accessing-attributes

attributeAt: aKey
Not commented.

attributeAt: aKey put: anObject
Not commented.

13.21.3 Iliad.ILContext: adding

addBuiltWidget: aWidget
Not commented.

13.22 Iliad.ILCookie

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-lib-HTTP

From Seaside 2.8

I represent a cookie, a piece of information that is stored on the client and read and writable by the server. I am basically a key/value pair of strings. You can never trust information in a cookie, the client is free to edit it. I model only a part of the full cookie specification.

For more information see RFC2965 (http://tools.ietf.org/html/rfc2965)

13.22.1 Iliad.ILCookie class: instance creation

key: keyString value: valueString
Not commented.

13.22.2 Iliad.ILCookie: accessing

expiry Answer 'expiry'.

expiry: anInteger
Not commented.

key Answer 'key'.

key: aString
Not commented.

path Not commented.

path: aString
Not commented.

value Answer 'value'.

value: aString
Not commented.

13.22.3 Iliad.ILCookie: api

expireIn: aDuration
Not commented.

valueWithExpiry
Not commented.

13.22.4 Iliad.ILCookie: comparing

= other Not commented.

hash Not commented.

13.22.5 Iliad.ILCookie: writing

writeOn: aStream
Not commented.

13.23 Iliad.ILCurrentBuildable

Defined in namespace Iliad
Superclass: Iliad.ILDynamicVariable
Category: Iliad-Core-Buildables

13.24 Iliad.ILCurrentContext

Defined in namespace Iliad
Superclass: Iliad.ILDynamicVariable
Category: Iliad-Core-Sessions

13.25 Iliad.ILDecorator

Defined in namespace Iliad
Superclass: Iliad.ILBuildable
Category: Iliad-Core-Buildables

I am a decorator for Widgets. I can be added to a widget by calling #decorateWith: from a widget. Subclasses can be used to modify the building process of a widget, or change its behavior

13.25.1 Iliad.ILDecorator class: instance creation

decoratee: aDecoratee

Not commented.

13.25.2 Iliad.ILDecorator: accessing

decoratee

Answer 'decoratee'.

decoratee: aDecoratee

Not commented.

widget Not commented.

13.25.3 Iliad.ILDecorator: building

contents

Not commented.

scripts Answer '#()'.

styles Answer '#()'.

updateHead: aHead

Not commented.

13.25.4 Iliad.ILDecorator: decorations

handleAnswer: anAnswer
Not commented.

removeDecorator: aDecorator
Not commented.

removeYourself
Not commented.

13.25.5 Iliad.ILDecorator: testing

isDelegator
Answer 'false'.

isGlobal
Answer 'false'.

13.26 Iliad.ILDelegator

Defined in namespace Iliad
Superclass: Iliad.ILDecorator
Category: Iliad-Core-Buildables

13.26.1 Iliad.ILDelegator: accessing

newRootElement
Not commented.

widget Answer 'widget'.

widget: aWidget
Not commented.

13.26.2 Iliad.ILDelegator: building

contents
Not commented.

updateHead: aHead
Not commented.

13.26.3 Iliad.ILDelegator: decorations

handleAnswer: anAnswer
Not commented.

13.26.4 Iliad.ILDelegator: testing

isDelegator
Answer 'true'.

isGlobal
Answer 'true'.

13.27 Iliad.ILDirectionElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.27.1 Iliad.ILDirectionElement: accessing

tag Answer ''bdo".

xmlLang: aString
Not commented.

13.27.2 Iliad.ILDirectionElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.27.3 Iliad.ILDirectionElement: acessing attributes

dir: aString
Not commented.

13.27.4 Iliad.ILDirectionElement: printing

beforePrintHtml

Not commented.

13.28 Iliad.ILDirectory

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-RequestHandlers

13.28.1 Iliad.ILDirectory: accessing

fileContentsFor: aFilename

This method's functionality should be implemented by subclasses of ILDirectory

13.29 Iliad.ILDiskDirectory

Defined in namespace Iliad
Superclass: Iliad.ILDirectory
Category: Iliad-Core-GST

An ILDiskDirectory allows files to be served by Iliad directly from directories of even archives like .star files. I shouldn't be used in production.

Usage example:

Iliad.ILFileHandler addDirectory: (Iliad.ILDiskDirectory new directory: (PackageLoader packageAt: 'SomePackage') directory / 'Public';

13.29.1 Iliad.ILDiskDirectory: accessing

directory

Answer 'directory'.

directory: aDirectory

Not commented.

fileContentsFor: aString

Not commented.

13.30 Iliad.ILDispatcher

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Dispatching

I am the entry point of requests. I dispatch them with the #dispatch: method to an ILApplicationHandler or a ILFileHandler.

Web server adapters should use the #dispatch: method on the current instance of ILDispatcher - returned by ILDispatcher class>>current - to handle requests, and wait for a ILResponseNotification to respond to them.

13.30.1 Iliad.ILDispatcher class: instance creation

current Not commented.

new This method should not be called for instances of this class.

13.30.2 Iliad.ILDispatcher: dispatching

dispatch: aRequest
Entry point of requests

13.30.3 Iliad.ILDispatcher: error-handling

withDispatchErrorHandling: aBlock
Catch dispatch errors

withErrorHandling: aBlock
Catch errors and use an ILErrorHandler to handle them

13.31 Iliad.ILDispatchError

Defined in namespace Iliad
Superclass: Grease.Grease.GRError
Category: Iliad-Core-Dispatching

13.32 Iliad.ILDivElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.32.1 Iliad.ILDivElement: accessing

tag Answer ‘’div”.

13.33 Iliad.ILDynamicVariable

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-Utilities

13.33.1 Iliad.ILDynamicVariable class: defaults

defaultValue
Answer ‘nil’.

13.33.2 Iliad.ILDynamicVariable class: evaluating

use: anObject during: aBlock
Not commented.

value Not commented.

13.34 Iliad.ILElement

Defined in namespace Iliad
Superclass: Iliad.ILComposite
Category: Iliad-Core-Elements

I am the abstract root class of the composite element hierarchy.

I know how to print myself in HTML format with the #printHtmlOn: method

13.34.1 Iliad.ILElement: accessing

attributeAt: akey
Not commented.

attributeAt: akey ifAbsent: aBlock
Not commented.

attributeAt: akey ifAbsentPut: aBlock
Not commented.

attributeAt: aKey put: aValue
Not commented.

attributes
Not commented.

contentType
This method's functionality should be implemented by subclasses of ILElement

tag Answer 'nil'.

13.34.2 Iliad.ILElement: adding-conveniance

text: aString
Not commented.

xml Not commented.

13.34.3 Iliad.ILElement: building

build: aBuildable
Not commented.

13.34.4 Iliad.ILElement: comparing

= anObject
Not commented.

hash Not commented.

13.34.5 Iliad.ILElement: converting

respondOn: aResponse
Not commented.

13.34.6 Iliad.ILElement: error handling

attributeError: key
Not commented.

doesNotUnderstandAttribute: aString
Not commented.

13.34.7 Iliad.ILElement: printing

afterPrintHtml
Answer the receiver.

beforePrintHtml
Answer the receiver.

printAttribute: anAttribute on: aStream
Not commented.

printCloseTagOn: aStream
Not commented.

printHtmlOn: aStream
Not commented.

printJsonOn: aStream
Not commented.

printOpenTagOn: aStream
Not commented.

13.35 Iliad.ILElementError

Defined in namespace Iliad
Superclass: Grease.Grease.GRError
Category: Iliad-Core-Elements

13.36 Iliad.ILEncoder

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-Utilities

13.36.1 Iliad.ILEncoder class: encoding

encodeCharacterForHTTP: aCharacter on: aStream
Not commented.

encodeForHTTP: aString on: aStream
Not commented.

encodeUrl: aString on: aStream
Not commented.

encodeUrlCharacter: aCharacter on: aStream
Not commented.

printUrl: aString encoded: aBoolean on: aStream
Not commented.

13.37 Iliad.ILErrorHandler

Defined in namespace Iliad
Superclass: Iliad.ILRequestHandler
Category: Iliad-Core-RequestHandlers

"""""""""""""""""""""""" " Error handling modes " """"""""""""""""""""""""

Error handlers can be in one of the following modes: deployment, verbose or debug. The default mode is verbose. You can switch between modes with class methods in the <accessing modes> protocol.

When errors occur, the framework with handle them differently depending on the application mode: - in deployment mode, it will respond an error 500; - in verbose mode, it will also respond an error 500, but with error details; - in debug mode, a debugger window will be opened on the error;

13.37.1 Iliad.ILErrorHandler class: accessing

mode Not commented.

mode: aSymbol
Not commented.

13.37.2 Iliad.ILErrorHandler class: accessing modes

debugMode
Not commented.

deploymentMode
Not commented.

verboseMode
Not commented.

13.37.3 Iliad.ILErrorHandler: accessing

error Answer ‘error’.

error: anError
Not commented.

mode Not commented.

newResponse
Not commented.

13.37.4 Iliad.ILErrorHandler: responding

produceDebugResponse
Not commented.

produceDeploymentResponse
Not commented.

produceResponse
Not commented.

produceVerboseResponse
Not commented.

13.37.5 Iliad.ILErrorHandler: testing

isDebugMode
Not commented.

isDeploymentMode
Not commented.

isVerboseMode
Not commented.

13.38 Iliad.ILFieldsetElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.38.1 Iliad.ILFieldsetElement: accessing

tag Answer ‘’fieldset”.

13.39 Iliad.ILFileHandler

Defined in namespace Iliad
Superclass: Iliad.ILRequestHandler
Category: Iliad-Core-RequestHandlers

13.39.1 Iliad.ILFileHandler class: acccessing

directories
Not commented.

13.39.2 Iliad.ILFileHandler class: accessing

addDirectory: aDirectory
Not commented.

defaultMimeType
Answer ''application/octet-stream''.

defaultMimeTypes
Not commented.

directories: aCollection
Not commented.

mimeTypeFor: aString
Not commented.

mimeTypes
Not commented.

13.39.3 Iliad.ILFileHandler class: defaults

initMimeTypes
Not commented.

13.39.4 Iliad.ILFileHandler class: testing

isBinary: aFilename
Not commented.

13.39.5 Iliad.ILFileHandler: accessing

directories
Not commented.

mimeTypeFor: aFilename
Not commented.

newResponse
Not commented.

13.39.6 Iliad.ILFileHandler: request handling

handleRequest

Not commented.

13.39.7 Iliad.ILFileHandler: responding

produceResponse

Not commented.

13.40 Iliad.ILFileProxy

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-lib-HTTP

13.40.1 Iliad.ILFileProxy class: instance creation

new Not commented.

13.40.2 Iliad.ILFileProxy: accessing

contentType

Answer ‘contentType’.

contentType: anObject

Not commented.

contents

Answer ‘contents’.

contents: anObject

Not commented.

createdTimestamp

Not commented.

filename

Answer ‘filename’.

filename: anObject
Not commented.

timestamps
Not commented.

timestampsAt: aSymbol
Not commented.

timestampsAt: aSymbol put: aTimestamp
Not commented.

13.40.3 Iliad.ILFileProxy: initialization

initialize
Not commented.

setCreatedTimestamp
Not commented.

13.40.4 Iliad.ILFileProxy: writing

writeToFile
Not commented.

13.41 Iliad.ILFormElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.41.1 Iliad.ILFormElement class: defaults

encodingType
Answer ''application/x-www-form-urlencoded".

multipartFormData
Answer ''multipart/form-data".

13.41.2 Iliad.ILFormElement: accessing

tag Answer ''form''.

url Not commented.

13.41.3 Iliad.ILFormElement: accessing attributes

accept: aString
Not commented.

acceptCharset: aString
Not commented.

acceptCharsets: aCollection
Not commented.

accepts: aCollection
Not commented.

beMultipart
Not commented.

enctype Not commented.

enctype: aString
Not commented.

multipart: aBoolean
Not commented.

useGet Not commented.

usePost Not commented.

13.41.4 Iliad.ILFormElement: accessing attributes-events

onReset: aString
Not commented.

onResetDo: aBlock
Not commented.

onSubmit: aString
Not commented.

onSubmitDo: aBlock
Not commented.

13.41.5 Iliad.ILFormElement: printing

beforePrintHtml

Not commented.

13.41.6 Iliad.ILFormElement: testing

isMultipart

Not commented.

13.42 Iliad.ILFormElementElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.42.1 Iliad.ILFormElementElement: accessing

accesskey: aCharacter

Not commented.

13.42.2 Iliad.ILFormElementElement: accessing attributes

action: aBlock

Not commented.

beSubmitOnChange

Not commented.

beSubmitOnClick

Not commented.

beSubmitOnEvent: aString

Not commented.

disabled

Not commented.

disabled: aBoolean
Not commented.

name Not commented.

name: aString
Not commented.

readonly: aBoolean
Not commented.

13.43 Iliad.ILHeadElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.43.1 Iliad.ILHeadElement: accessing

tag Answer ''head".

13.43.2 Iliad.ILHeadElement: accessing attributes

profile: aString
Not commented.

profiles: aCollection
Not commented.

title Not commented.

title: aString
Not commented.

13.43.3 Iliad.ILHeadElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.43.4 Iliad.ILHeadElement: adding-conveniance

style Not commented.

style: aString
Not commented.

13.43.5 Iliad.ILHeadElement: printing

beforePrintHtml
Not commented.

13.44 Iliad.ILHeadingElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.44.1 Iliad.ILHeadingElement: accessing

level Answer 'level'.

level: anInteger
Not commented.

tag Not commented.

13.44.2 Iliad.ILHeadingElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.45 Iliad.ILHorizontalRuleElement

Defined in namespace Iliad
Superclass: Iliad.ILClosingElement
Category: Iliad-Core-XHTMLElements

13.45.1 Iliad.ILHorizontalRuleElement: accessing

tag Answer ‘’hr”.

13.45.2 Iliad.ILHorizontalRuleElement: accessing attributes-events

onEvent: anEvent add: aString
Not commented.

13.46 Iliad.ILHTMLBuilderElement

Defined in namespace Iliad
Superclass: Iliad.ILElement
Category: Iliad-Core-XHTMLElements

13.46.1 Iliad.ILHTMLBuilderElement: accessing

class: aString
Not commented.

contentType
Not commented.

if Not commented.

13.46.2 Iliad.ILHTMLBuilderElement: accessing attributes-events

onChange: aString
Not commented.

onChangeDo: aBlock
Not commented.

13.46.3 Iliad.ILHTMLBuilderElement: accessing atibutes

style: aString
Not commented.

13.46.4 Iliad.ILHTMLBuilderElement: accessing attributes

cssClass
Not commented.

cssClass: aString
Not commented.

id Not commented.

id: aString
Not commented.

lang: aString
Not commented.

style Not commented.

title Not commented.

title: aString
Not commented.

xmlLang: aString
Not commented.

13.46.5 Iliad.ILHTMLBuilderElement: accessing attributes-events

onBlur: aString
Not commented.

onBlurDo: aBlock
Not commented.

onClick: aString
Not commented.

onClickDo: aBlock
Not commented.

onDoubleClick: aString
Not commented.

onDoubleClickDo: aBlock
Not commented.

onEvent: aString add: anotherString
Not commented.

onEvent: aString do: aBlock
Not commented.

onFocus: aString
Not commented.

onKeyDown: aString
Not commented.

onKeyPress: aString
Not commented.

onKeyUp: aString
Not commented.

onMouseOut: aString
Not commented.

onMouseOver: aString
Not commented.

onSelect: aString
Not commented.

13.46.6 Iliad.ILHTMLBuilderElement: adding-conveenience

link Not commented.

13.46.7 Iliad.ILHTMLBuilderElement: adding-conveniance

a Not commented.

area Not commented.

b Not commented.

b: aString
Not commented.

bdo Not commented.

big Not commented.

big: aString
Not commented.

blockquote
Not commented.

blockquote: aString
Not commented.

br Not commented.

button Not commented.

checkbox
Not commented.

cite Not commented.

cite: aString
Not commented.

code Not commented.

code: aString
Not commented.

dd Not commented.

dfn Not commented.

div Not commented.

dl Not commented.

dt Not commented.

em Not commented.

em: aString
Not commented.

favicon Not commented.

favicon: aString
Not commented.

fieldset Not commented.

file Not commented.

form Not commented.

h Not commented.

h1 Not commented.

h1: aString
Not commented.

h2 Not commented.

h2: aString
Not commented.

h3 Not commented.

h3: aString
Not commented.

h4 Not commented.

h4: aString
Not commented.

h5 Not commented.

h5: aString
Not commented.

h6 Not commented.

h6: aString
Not commented.

hr Not commented.

html: aString
Not commented.

i Not commented.

i: aString
Not commented.

img Not commented.

img: aString
Not commented.

input Not commented.

javascript
Not commented.

label Not commented.

legend Not commented.

legend: aString
Not commented.

li Not commented.

map Not commented.

meta Not commented.

object Not commented.

ol Not commented.

optgroup
Not commented.

option Not commented.

p Not commented.

param Not commented.

password
Not commented.

pre Not commented.

pre: aString
Not commented.

quote Not commented.

quote: aString
Not commented.

radio Not commented.

reset Not commented.

script Not commented.

script: aString
Not commented.

select Not commented.

small Not commented.

small: aString
Not commented.

space Not commented.

span Not commented.

strong Not commented.

strong: aString
Not commented.

stylesheet
Not commented.

submit Not commented.

subscript
Not commented.

subscript: aString
Not commented.

superscript
Not commented.

superscript: aString
Not commented.

table Not commented.

tbody Not commented.

td Not commented.

textarea
Not commented.

tfoot Not commented.

th Not commented.

thead Not commented.

tr Not commented.

tt Not commented.

ul Not commented.

var Not commented.

var: aString
Not commented.

13.47 Iliad.ILHtmlElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.47.1 Iliad.ILHtmlElement: accessing

beHtml5
Not commented.

beXhtml10Strict
Not commented.

beXhtml10Transitional
Not commented.

beXhtml11
Not commented.

doctype Answer ‘doctype’.

doctype: aString
Not commented.

setXmlTag
Not commented.

tag Answer ‘’html”.

xmlTag Answer ‘xmlTag’.

xmlTag: aString
Not commented.

13.47.2 Iliad.ILHtmlElement: accessing attributes

version: aNumber
Not commented.

xmlns Not commented.

xmlns: aString
Not commented.

13.47.3 Iliad.ILHtmlElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.47.4 Iliad.ILHtmlElement: defaults

defaultXmlns
Answer ''http://www.w3.org/1999/xhtml''.

13.47.5 Iliad.ILHtmlElement: printing

beforePrintHtml
Not commented.

printHtmlOn: aStream
Not commented.

13.47.6 Iliad.ILHtmlElement: testing

hasXmlTag
Not commented.

13.48 Iliad.ILHTMLPage

Defined in namespace Iliad
Superclass: Iliad.ILBuildable
Category: Iliad-Core-Buildables

13.48.1 Iliad.ILHTMLPage: accessing attributes

body Not commented.

head Not commented.

html Not commented.

13.48.2 Iliad.ILHTMLPage: building

build Not commented.

13.49 Iliad.ILId

Defined in namespace Iliad
Superclass: ByteArray
Category: Iliad-Core-Utilities

13.49.1 Iliad.ILId class: defaults

defaultSize
Answer '12'.

13.49.2 Iliad.ILId class: instance creation

new Not commented.

new: anInteger
Not commented.

13.49.3 Iliad.ILId: accessing

table Not commented.

13.49.4 Iliad.ILId: initialization

initialize
Not commented.

13.49.5 Iliad.ILId: printing

printOn: aStream
Not commented.

13.50 Iliad.ILImageElement

Defined in namespace Iliad
Superclass: Iliad.ILClosingElement
Category: Iliad-Core-XHTMLElements

13.50.1 Iliad.ILImageElement: accessing

tag Answer ''img".

13.50.2 Iliad.ILImageElement: accessing attributes

alt: aString
Not commented.

height: anInteger
Not commented.

ismap Not commented.

longdesc: aString
Not commented.

src: aString
Not commented.

src: src alt: alt
Not commented.

usemap: aString
Not commented.

width: anInteger
Not commented.

13.50.3 Iliad.ILImageElement: printing

beforePrintHtml
Not commented.

13.51 Iliad.ILInformationWidget

Defined in namespace Iliad
Superclass: Iliad.ILWidget
Category: Iliad-Core-Buildables

13.51.1 Iliad.ILInformationWidget: accessing

informationString
Answer 'informationString'.

informationString: aString
Not commented.

13.51.2 Iliad.ILInformationWidget: building

contents
Not commented.

13.52 Iliad.ILInputElement

Defined in namespace Iliad
Superclass: Iliad.ILFormElementElement
Category: Iliad-Core-XHTMLElements

13.52.1 Iliad.ILInputElement: accessing

tag Answer ''input''.

13.52.2 Iliad.ILInputElement: accessing attributes

accept: aString
Not commented.

accepts: aCollection
Not commented.

alt: aString
Not commented.

ismap Not commented.

maxlength: anInteger
Not commented.

size: anInteger
Not commented.

usemap: aString
Not commented.

value: aString
Not commented.

13.52.3 Iliad.ILInputElement: accessing attributes-types

beFile Not commented.

beHidden
Not commented.

beImage
Not commented.

bePassword
Not commented.

beReset
Not commented.

beSubmit
Not commented.

beText Not commented.

type: aString
Not commented.

13.52.4 Iliad.ILInputElement: printing

beforePrintHtml
Not commented.

printHtmlOn: aStream
Not commented.

13.53 Iliad.ILJson

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-lib-JSON

This class reads and writes JSON format data - strings, numbers, boolean, nil, arrays and dictionaries. See http://www.crockford.com/JSON/index.html. It has been extended with syntax for invoking a prearranged list of constructors on read objects.

13.53.1 Iliad.ILJson: accessing

ctorMap
Answer ‘ctorMap’.

ctorMap: m
Not commented.

stream Answer the value of stream

stream: anObject
Set the value of stream

13.53.2 Iliad.ILJson: parsing

readAny
This is the main entry point for the JSON parser. See also readFrom: on the class side.

readFrom: aStream
Not commented.

13.54 Iliad.ILJsonHandler

Defined in namespace Iliad
Superclass: Iliad.ILRequestHandler
Category: Iliad-Core-RequestHandlers

13.54.1 Iliad.ILJsonHandler: initialization

initialize
Not commented.

13.54.2 Iliad.ILJsonHandler: responding

produceResponse
Not commented.

13.54.3 Iliad.ILJsonHandler: testing

shouldRedirect
Not commented.

shouldUpdateApplication
Not commented.

13.55 Iliad.ILJsonObject

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-lib-JSON

13.55.1 Iliad.ILJsonObject: accessing

at: key Not commented.

at: key ifAbsent: aBlock
Not commented.

at: key put: value
Not commented.

properties
Answer 'properties'.

13.55.2 Iliad.ILJsonObject: error handling

doesNotUnderstand: aMessage
Not commented.

13.55.3 Iliad.ILJsonObject: initialization

initialize
Not commented.

13.56 Iliad.ILJsonSyntaxError

Defined in namespace Iliad
Superclass: Grease.Grease.GRError
Category: Iliad-Core-lib-JSON
Class Json signals instances of me when an input stream contains invalid JSON input.

13.57 Iliad.ILLabelElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.57.1 Iliad.ILLabelElement: accessing

tag Answer ''label".

13.57.2 Iliad.ILLabelElement: accessing attributes

accesskey: aCharacter
Not commented.

for: aString
Not commented.

13.58 Iliad.ILLegendElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.58.1 Iliad.ILLegendElement: accessing

tag Answer ''legend".

13.58.2 Iliad.ILLegendElement: accessing attributes

accesskey: aCharacter
Not commented.

13.59 Iliad.ILLinkableElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.59.1 Iliad.ILLinkableElement: accessing attributes

accesskey: aCharacter
Not commented.

charset: aString
Not commented.

contentType: aString
Not commented.

href: aString
Not commented.

hreflang: aString
Not commented.

linkToApplication: anApplicationClass
Not commented.

linkToApplication: anApplicationClass controller: aString
Not commented.

linkToLocal: aString
Not commented.

target: aString
Not commented.

type: aString
Not commented.

13.59.2 Iliad.ILLinkableElement: accessing attributes-relationships

beAlternate
Not commented.

beAppendix
Not commented.

beBookmark
Not commented.

beChapter
Not commented.

beContents
Not commented.

beCopyright
Not commented.

beFavicon
Not commented.

beGlossary
Not commented.

beHelp Not commented.

beIndex
Not commented.

beNext Not commented.

bePrev Not commented.

bePrevious
Not commented.

beRss Not commented.

beSection
Not commented.

beStart Not commented.

beStylesheet
Not commented.

beSubsection
Not commented.

rel: aString
Not commented.

13.59.3 Iliad.ILLinkableElement: accessing attributes-reverse links

fromAlternate
Not commented.

fromAppendix
Not commented.

fromBookmark
Not commented.

fromChapter
Not commented.

fromContents
Not commented.

fromCopyright
Not commented.

fromGlossary
Not commented.

fromHelp
Not commented.

fromIndex
Not commented.

fromNext
Not commented.

fromPrev
Not commented.

fromPrevious
Not commented.

fromSection
Not commented.

fromStart
Not commented.

fromStylesheet
Not commented.

fromSubsection
Not commented.

rev: aString
Not commented.

13.60 Iliad.ILLinkElement

Defined in namespace Iliad
Superclass: Iliad.ILLinkableElement
Category: Iliad-Core-XHTMLElements

13.60.1 Iliad.ILLinkElement: accessing

tag Answer ''link".

13.60.2 Iliad.ILLinkElement: accessing attributes

allMedia
Not commented.

auralMedia
Not commented.

brailleMedia
Not commented.

handheldMedia
Not commented.

media: aString
Not commented.

printMedia
Not commented.

projectionMedia
Not commented.

screenMedia
Not commented.

ttyMedia
Not commented.

tvMedia
Not commented.

type: aString
Not commented.

13.60.3 Iliad.ILLinkElement: printing

printHtmlOn: aStream
Not commented.

13.61 Iliad.ILListElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.61.1 Iliad.ILListElement: accessing

beOrdered
Not commented.

beUnordered
Not commented.

tag Answer ‘tag’.

tag: aString
Not commented.

13.62 Iliad.ILListItemElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.62.1 Iliad.ILListItemElement: accessing

tag Answer ‘’li”.

13.63 Iliad.ILMapElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.63.1 Iliad.ILMapElement: accessing

tag Answer ''map".

13.63.2 Iliad.ILMapElement: accessing attributes

classes: aCollection
Not commented.

13.64 Iliad.ILMemoryDirectory

Defined in namespace Iliad
Superclass: Iliad.ILDirectory
Category: Iliad-Core-RequestHandlers

13.64.1 Iliad.ILMemoryDirectory class: maintenance

addAllFilesIn: aPathString
Answer the receiver.

addFileAt: aPath
Answer the receiver.

13.64.2 Iliad.ILMemoryDirectory: accessing

fileContentsFor: aString
Not commented.

fileSelectors
Not commented.

path answer the base path of the memory directory

13.64.3 Iliad.ILMemoryDirectory: maintenence

deployFiles

Write to disk the files that the receiver use to serve as methods. The files are stored in a subfolder named like the classname of the receiver in a subfolder of Smalltalk image folder.

removeFile: aFilename

Not commented.

13.64.4 Iliad.ILMemoryDirectory: testing

isFileSelector: aSelector

Only methods in 'files' protocol are allowed to be served as files

13.65 Iliad.ILMetaElement

Defined in namespace Iliad
Superclass: Iliad.ILClosingElement
Category: Iliad-Core-XHTMLElements

13.65.1 Iliad.ILMetaElement: accessing

tag Answer ''meta".

13.65.2 Iliad.ILMetaElement: accessing attributes

class: aString

Not commented.

classes: aCollection

Not commented.

content: aString

Not commented.

contentType

Not commented.

expires Not commented.

httpEquiv: aString
Not commented.

name: aString
Not commented.

refresh Not commented.

scheme: aString
Not commented.

setCookie
Not commented.

title: aString
Not commented.

13.65.3 Iliad.ILMetaElement: accessing attributes-events

onEvent: event add: aString
Not commented.

13.65.4 Iliad.ILMetaElement: printing

beforePrintHtml
Not commented.

13.66 Iliad.ILModelProxy

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-Utilities

13.66.1 Iliad.ILModelProxy class: instance creation

on: anObject
Not commented.

13.66.2 Iliad.ILModelProxy: accessing

model Answer ‘model’.

13.66.3 Iliad.ILModelProxy: forwarding

commit Not commented.

doesNotUnderstand: aMessage
Not commented.

13.66.4 Iliad.ILModelProxy: initializing

setModel: anObject
Not commented.

13.67 Iliad.ILNotFoundHandler

Defined in namespace Iliad
Superclass: Iliad.ILRequestHandler
Category: Iliad-Core-RequestHandlers

13.67.1 Iliad.ILNotFoundHandler: responding

produceResponse
Not commented.

13.68 Iliad.ILObject

Defined in namespace Iliad
Superclass: Grease.Grease.GRObject
Category: Iliad-Core-Utilities
ILObject is the base class for all Iliad classes. It guarantees that #initialize is send upon object creation. Additionally it provides convienience methods for accessing the current context, session and application.

13.68.1 Iliad.ILObject: accessing

application
Not commented.

context Not commented.

request Not commented.

session Not commented.

urlBuilder
Not commented.

13.69 Iliad.ILObjectElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.69.1 Iliad.ILObjectElement: accessing

tag Answer ''object".

13.69.2 Iliad.ILObjectElement: accessing attributes

archive: aString
Not commented.

archives: aCollection
Not commented.

classid: aString
Not commented.

codebase: aString
Not commented.

codetype: aString
Not commented.

data: aString
Not commented.

declare Not commented.

declareOnly
Not commented.

height: anInteger
Not commented.

name: aString
Not commented.

standby: aString
Not commented.

tabindex: anInteger
Not commented.

type: aString
Not commented.

usemap: aString
Not commented.

width: anInteger
Not commented.

13.70 Iliad.ILOptionElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.70.1 Iliad.ILOptionElement: accessing

tag Answer ‘’option”.

13.70.2 Iliad.ILOptionElement: accessing attributes

disabled
Not commented.

disabled: aBoolean
Not commented.

label: aString
Not commented.

selected Not commented.

selected: aBoolean
Not commented.

value: aString
Not commented.

13.70.3 Iliad.ILOptionElement: accessing-attributes

action: aBlock
Not commented.

13.71 Iliad.ILOptionGroupElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.71.1 Iliad.ILOptionGroupElement: accessing

tag Answer ''optgroup''.

13.71.2 Iliad.ILOptionGroupElement: accessing attributes

disabled
Not commented.

label: aString
Not commented.

13.72 Iliad.ILParagraphElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.72.1 Iliad.ILParagraphElement: accessing

tag Answer ‘’p”.

13.73 Iliad.ILParameterElement

Defined in namespace Iliad
Superclass: Iliad.ILClosingElement
Category: Iliad-Core-XHTMLElements

13.73.1 Iliad.ILParameterElement: accessing

tag Answer ‘’param”.

13.73.2 Iliad.ILParameterElement: accessing attributes

beData Not commented.

beObject
Not commented.

beReference
Not commented.

name: aString
Not commented.

type: aString
Not commented.

value: aString
Not commented.

valuetype: aString
Not commented.

13.73.3 Iliad.ILParameterElement: adding

add: anElement

Not commented.

13.73.4 Iliad.ILParameterElement: printing

beforePrintHtml

Not commented.

13.74 Iliad.ILPluggableWidget

Defined in namespace Iliad
Superclass: Iliad.ILWidget
Category: Iliad-Core-Buildables

13.74.1 Iliad.ILPluggableWidget: accessing

contentsBlock

Not commented.

contentsBlock: aBlock

Not commented.

13.74.2 Iliad.ILPluggableWidget: building

contents

Not commented.

13.75 Iliad.ILPrependDelegator

Defined in namespace Iliad
Superclass: Iliad.ILDelegator
Category: Iliad-Core-Buildables

13.75.1 Iliad.ILPrependDelegator: building

contents
Not commented.

13.76 Iliad.ILProfiler

Defined in namespace Iliad
Superclass: Iliad.ILDecorator
Category: Iliad-Core-Buildables

13.76.1 Iliad.ILProfiler: building

contents
Not commented.

13.77 Iliad.ILRadioButtonElement

Defined in namespace Iliad
Superclass: Iliad.ILFormElementElement
Category: Iliad-Core-XHTMLElements

13.77.1 Iliad.ILRadioButtonElement: accessing

tag Answer ‘’input”.

13.77.2 Iliad.ILRadioButtonElement: accessing attributes

action: aBlock
Not commented.

beSelected
Not commented.

selected: aBoolean
Not commented.

value: aString
Not commented.

13.77.3 Iliad.ILRadioButtonElement: printing

beforePrintHtml
Not commented.

13.78 Iliad.ILRawHtmlElement

Defined in namespace Iliad
Superclass: Iliad.ILElement
Category: Iliad-Core-XHTMLElements

13.78.1 Iliad.ILRawHtmlElement: accessing

contents
Not commented.

contents: aString
Not commented.

13.78.2 Iliad.ILRawHtmlElement: printing

printHtmlOn: aStream
Not commented.

13.79 Iliad.ILRedirectHandler

Defined in namespace Iliad
Superclass: Iliad.ILRequestHandler
Category: Iliad-Core-RequestHandlers

13.79.1 Iliad.ILRedirectHandler: responding

produceResponse
Not commented.

13.79.2 Iliad.ILRedirectHandler: testing

shouldRespondInJson
Not commented.

13.80 Iliad.ILRequest

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-lib-HTTP
From Seaside 2.8.

I am a server independent http request object.

13.80.1 Iliad.ILRequest class: instance creation

new Not commented.

13.80.2 Iliad.ILRequest: accessing

actionField
Not commented.

ajaxUploadField
Not commented.

at: key Not commented.

at: key ifAbsent: aBlock
Not commented.

cookies Answer ‘cookies’.

cookies: anObject
Not commented.

fields Answer ‘fields’.

fields: anObject
Not commented.

hashLocationField
Not commented.

headerAt: aKey
Not commented.

headerAt: aKey ifAbsent: aBlock
Not commented.

headers Answer ‘headers’.

headers: aDictionary
Not commented.

jsonField
Not commented.

method Answer ‘method’.

method: aString
Not commented.

nativeRequest
Answer ‘nativeRequest’.

nativeRequest: aRequest
Not commented.

password
Not commented.

sessionField
Not commented.

stateField
Not commented.

url Answer ‘url’.

url: anUrl
Not commented.

user Not commented.

userAgent
Not commented.

13.80.3 Iliad.ILRequest: initialization

initialize
Not commented.

13.80.4 Iliad.ILRequest: testing

hasCookies
Not commented.

isAjaxRequest
Not commented.

isGet Not commented.

isPost Not commented.

isPrefetch
Not commented.

isPut Not commented.

isTypeOfRequestForJson
Not commented.

isTypeOfRequestForRedirect
Not commented.

13.81 Iliad.ILRequestHandler

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-RequestHandlers
I implement the basic behavior needed to handle requests.

My sublcasses must override #newResponse to handle requests, and most likely #produceResponse. The current handled request is answered by the #request method inherited from ILObject

13.81.1 Iliad.ILRequestHandler: request handling

handleRequest
Not commented.

13.81.2 Iliad.ILRequestHandler: responding

produceResponse
Not commented.

13.82 Iliad.ILResponse

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-lib-HTTP
From Seaside 2.8

I am a server independent http response object.

13.82.1 Iliad.ILResponse class: instance creation

forbidden
Not commented.

new Not commented.

notFound
Not commented.

ok Not commented.

redirect Not commented.

13.82.2 Iliad.ILResponse: accessing

addCookie: aCookie
Not commented.

contentType
Answer ‘contentType’.

contentType: mimeTypeString
Not commented.

contents
Not commented.

contents: aStream
Not commented.

cookieAt: key put: value
Not commented.

cookies Not commented.

headerAt: key put: value
Not commented.

headers Not commented.

initialize
Not commented.

release Not commented.

status Answer 'status'.

status: anInteger
Not commented.

stream Answer 'stream'.

stream: aStream
Not commented.

13.82.3 Iliad.ILResponse: convenience

attachmentWithFileName: aString
Not commented.

basicAuthenticationRealm: aString
Not commented.

redirectTo: aString
Not commented.

13.82.4 Iliad.ILResponse: printing

printOn: aStream
Not commented.

13.82.5 Iliad.ILResponse: status

authenticationFailed
Not commented.

forbidden
Not commented.

internalError
Not commented.

notFound
Not commented.

ok Not commented.

redirect Not commented.

13.82.6 Iliad.ILResponse: streaming

nextPut: aCharacter
Not commented.

nextPutAll: aString
Not commented.

space Not commented.

13.83 Iliad.ILResponseNotification

Defined in namespace Iliad
Superclass: Notification
Category: Iliad-Core-RequestHandlers

13.83.1 Iliad.ILResponseNotification: accessing

response
Answer ‘response’.

response: aResponse
Not commented.

13.84 Iliad.ILRoute

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Dispatching

I represent a route, used to dispatch a request. My path is a collection of string, representing each part of the path of an url, and can be streamed with methods in the <streaming> protocol.

13.84.1 Iliad.ILRoute class: accessing

delimiters

Not commented.

13.84.2 Iliad.ILRoute class: instance creation

path: aCollection

Not commented.

13.84.3 Iliad.ILRoute: accessing

basePath

Not commented.

delimiter

Not commented.

delimiters

Not commented.

path Not commented.

13.84.4 Iliad.ILRoute: converting

pathString

Not commented.

uriString

Not commented.

13.84.5 Iliad.ILRoute: initialization

initializeWithPath: aCollection
Not commented.

13.84.6 Iliad.ILRoute: iterating

currentPath
Return an absolute url of the current streamed path separated with delimiters, like this: /foo/bar/baz

13.84.7 Iliad.ILRoute: printing

printOn: aStream
Not commented.

13.84.8 Iliad.ILRoute: streaming

atEnd Not commented.

next Not commented.

peek Not commented.

position Not commented.

position: anInteger
Not commented.

reset Not commented.

13.85 Iliad.ILRouter

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Dispatching

13.85.1 Iliad.ILRouter: accessing

application
Answer 'application'.

controller
Answer 'controller'.

hashRoute
Not commented.

route Not commented.

13.85.2 Iliad.ILRouter: dispatching

dispatchRequest
Not commented.

13.85.3 Iliad.ILRouter: initialization

initialize
Not commented.

13.85.4 Iliad.ILRouter: testing

shouldRedirect
Not commented.

13.85.5 Iliad.ILRouter: updating

updateApplicationFromRoute
Not commented.

13.86 Iliad.ILRubyTextElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.86.1 Iliad.ILRubyTextElement: accessing

tag Answer ''rt".

13.86.2 Iliad.ILRubyTextElement: accessing attributes

rbspan: anInteger
Not commented.

13.87 Iliad.ILScriptElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.87.1 Iliad.ILScriptElement: accessing

contents
Not commented.

contents: aString
Not commented.

tag Answer ''script".

13.87.2 Iliad.ILScriptElement: accessing attributes

beJavascript
Not commented.

charset: aString
Not commented.

defer Not commented.

language: aString
Not commented.

src: aString
Not commented.

type: aString
Not commented.

13.87.3 Iliad.ILScriptElement: printing

beforePrintHtml
Not commented.

printHtmlOn: aStream
do not encode contents!!

13.88 Iliad.ILSelectElement

Defined in namespace Iliad
Superclass: Iliad.ILFormElementElement
Category: Iliad-Core-XHTMLElements

13.88.1 Iliad.ILSelectElement: accessing

tag Answer ‘’select”.

13.88.2 Iliad.ILSelectElement: accessing attributes

action: aBlock
This method should not be called for instances of this class.

beMultiple
Not commented.

size: anInteger
Not commented.

13.88.3 Iliad.ILSelectElement: initialize release

initialize
Not commented.

13.89 Iliad.ILSequence

Defined in namespace Iliad
Superclass: Iliad.ILWidget
Category: Iliad-Core-Buildables

13.89.1 Iliad.ILSequence: building

build Not commented.

contents
This method should not be called for instances of this class.

13.89.2 Iliad.ILSequence: control flow

restart Not commented.

start Answer the receiver.

13.89.3 Iliad.ILSequence: testing

shouldStart
Not commented.

13.90 Iliad.ILSession

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions
I represent a session in Iliad. I persist as long as I am active (i.e. an user interacts with an application). When I am inactive, I expire after a timeout set by #expirySeconds. I also store actions and applications

13.90.1 Iliad.ILSession: accessing

applications
Not commented.

dirtyWidgets
Not commented.

encoding
Not commented.

id Answer 'id'.

id: anObject
Not commented.

nextId Not commented.

previousStateRegistry
Not commented.

route Not commented.

sessionManager
Not commented.

stateRegistry
Not commented.

13.90.2 Iliad.ILSession: accessing others

otherAt: aKey
Not commented.

otherAt: aKey ifAbsentPut: aBlock
Not commented.

otherAt: aKey put: anObject
Not commented.

others Not commented.

13.90.3 Iliad.ILSession: accessing preferences

charset Not commented.

charset: aString
Not commented.

expirySeconds
Not commented.

expirySeconds: anInteger
Not commented.

language
Not commented.

language: aSymbol
Not commented.

preferenceAt: aSymbol
Not commented.

preferenceAt: aSymbol ifAbsentPut: aBlock
Not commented.

preferenceAt: aSymbol put: anObject
Not commented.

preferences
Not commented.

refreshOnBacktrack
Not commented.

refreshOnBacktrack: aBoolean
Not commented.

useCookies
Not commented.

useCookies: aBoolean
Not commented.

13.90.4 Iliad.ILSession: accessing timestamps

createdTimestamp
Not commented.

modifiedTimestamp
Not commented.

setCreatedTimestamp
Not commented.

setModifiedTimestamp
Not commented.

timestampAt: aSymbol
Not commented.

timestampAt: aSymbol ifAbsentPut: aBlock
Not commented.

timestampAt: aSymbol put: anObject
Not commented.

timestamps
Not commented.

13.90.5 Iliad.ILSession: actions

actionAt: aKeyString
Not commented.

actionFor: aBlock
Not commented.

actionRegistry
Not commented.

evaluateActionKey: aString
Not commented.

registerAction: anAction
Not commented.

registerActionFor: aBlock
Not commented.

13.90.6 Iliad.ILSession: cleaning

clearActionRegistry
Not commented.

clearRedirectUrl
Not commented.

clearStateRegistries
Not commented.

13.90.7 Iliad.ILSession: defaults

defaultExpirySeconds
Answer ‘1800’.

defaultLanguage
Answer ‘’en”.

13.90.8 Iliad.ILSession: events

onExpire
Not commented.

13.90.9 Iliad.ILSession: initialization

initialize
Not commented.

13.90.10 Iliad.ILSession: redirection

redirect Not commented.

redirectTo: anUrlString
Not commented.

redirectToIndex
Not commented.

redirectToLocal: anUrlString
Not commented.

redirectUrl

Answer ‘redirectUrl’.

redirectUrl: aString

Not commented.

13.90.11 Iliad.ILSession: states

newStateRegistry

Not commented.

stateRegistries

Not commented.

stateRegistryAt: aString

Not commented.

13.90.12 Iliad.ILSession: testing

isExpired

Not commented.

shouldUseSessionField

Not commented.

13.91 Iliad.ILSessionManager

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions

I am responsible for managing, creating and removing sessions. The class method #current answers my only instance.

You shouldn’t subclass me. Instead, you can change my preferences with methods in the <accessing preferences> protocol.

In addition, do not try to access the current session from here, use ILContext>>session instead.

– maintenance –

To remove all sessions, even not expired ones, call #removeAllSessions. To remove all expired sessions, call #removeExpiredSessions

13.91.1 Iliad.ILSessionManager class: instance creation

current Not commented.

new This method should not be called for instances of this class.

13.91.2 Iliad.ILSessionManager: accessing

expiredSessions
Not commented.

sessions Not commented.

13.91.3 Iliad.ILSessionManager: accessing preferences

cookieName
Not commented.

cookieName: aString
Not commented.

preferenceAt: aSymbol
Not commented.

preferenceAt: aSymbol ifAbsentPut: aBlock
Not commented.

preferenceAt: aSymbol put: anObject
Not commented.

preferences
Not commented.

sessionClass
Not commented.

sessionClass: aSessionClass
Not commented.

13.91.4 Iliad.ILSessionManager: adding-removing

addSession: aSession
Not commented.

removeSession: aSession
Not commented.

sessionFor: aRequest
Answer the according session for <aRequest>. the session id may be found from <aRequest> cookies or <aRequest> fields. If no session is found, create and return a new one

13.91.5 Iliad.ILSessionManager: enumerating

sessionsDo: aBlock
Not commented.

13.91.6 Iliad.ILSessionManager: initialization

initialize
Not commented.

13.91.7 Iliad.ILSessionManager: testing

hasSession: aSession
Not commented.

shouldRemoveExpiredSessions
Do not remove all expired sessions for each request

13.92 Iliad.ILSpanElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.92.1 Iliad.ILSpanElement: accessing

tag Answer ''span".

13.93 Iliad.ILStateRegistry

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions

13.93.1 Iliad.ILStateRegistry class: instance creation

fromRegistry: aStateRegistry
Not commented.

13.93.2 Iliad.ILStateRegistry: accessing

key Not commented.

setStates: aDictionary
Not commented.

stateAt: aWidget
Not commented.

states Not commented.

widgets Not commented.

13.93.3 Iliad.ILStateRegistry: states

dirtyChildrenOf: aBuildable
Not commented.

dirtyWidgets
Answer all widgets which state has changed

register: aWidget
Not commented.

unregister: aWidget
Not commented.

unregisterAllWidgets
Not commented.

13.93.4 Iliad.ILStateRegistry: testing

isWidgetDirty: aWidget
Not commented.

13.94 Iliad.ILTableBodyElement

Defined in namespace Iliad
Superclass: Iliad.ILTableElementElement
Category: Iliad-Core-XHTMLElements

13.94.1 Iliad.ILTableBodyElement: accessing

tag Answer ''tbody''.

13.95 Iliad.ILTableCellElement

Defined in namespace Iliad
Superclass: Iliad.ILTableElementElement
Category: Iliad-Core-XHTMLElements

13.95.1 Iliad.ILTableCellElement: accessing attributes-table

abbr: aString
Not commented.

axis: aString
Not commented.

colScope
Not commented.

colgroupScope
Not commented.

colspan: anInteger
Not commented.

headers: aCollection
Not commented.

rowScope
Not commented.

rowgroupScope
Not commented.

rowspan: anInteger
Not commented.

scope: aString
Not commented.

13.96 Iliad.ILTableDataElement

Defined in namespace Iliad
Superclass: Iliad.ILTableCellElement
Category: Iliad-Core-XHTMLElements

13.96.1 Iliad.ILTableDataElement: accessing

tag Answer ''td''.

13.97 Iliad.ILTableElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.97.1 Iliad.ILTableElement: accessing

tag Answer ''table".

13.97.2 Iliad.ILTableElement: accessing attributes

aboveFrame
Not commented.

allRules Not commented.

belowFrame
Not commented.

border: anInteger
Not commented.

borderFrame
Not commented.

boxFrame
Not commented.

cellpadding: anInteger
Not commented.

cellspacing: anInteger
Not commented.

colsRules
Not commented.

frame: aString
Not commented.

groupRules
Not commented.

hsidesFrame
Not commented.

lhsFrame
Not commented.

noRules
Not commented.

rhsFrame
Not commented.

rowsRules
Not commented.

rules: aString
Not commented.

summary: aString
Not commented.

voidFrame
Not commented.

vsidesFrame
Not commented.

width: anInteger
Not commented.

13.98 Iliad.ILTableElementElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.98.1 Iliad.ILTableElementElement: accessing attributes-table

align: aString
Not commented.

baselineValign
Not commented.

bottomValign
Not commented.

centerAlign
Not commented.

character: aCharacter
Not commented.

characterAlign
Not commented.

charoff: anInteger
Not commented.

justifyAlign
Not commented.

leftAlign
Not commented.

middleValign
Not commented.

rightAlign
Not commented.

topValign
Not commented.

valign: aString
Not commented.

13.99 Iliad.ILTableFootElement

Defined in namespace Iliad
Superclass: Iliad.ILTableElementElement
Category: Iliad-Core-XHTMLElements

13.99.1 Iliad.ILTableFootElement: accessing

tag Answer ''tfoot".

13.100 Iliad.ILTableHeadElement

Defined in namespace Iliad
Superclass: Iliad.ILTableElementElement
Category: Iliad-Core-XHTMLElements

13.100.1 Iliad.ILTableHeadElement: accessing

tag Answer ''thead".

13.101 Iliad.ILTableHeaderElement

Defined in namespace Iliad
Superclass: Iliad.ILTableCellElement
Category: Iliad-Core-XHTMLElements

13.101.1 Iliad.ILTableHeaderElement: accessing

tag Answer ''th".

13.102 Iliad.ILTableRowElement

Defined in namespace Iliad
Superclass: Iliad.ILTableCellElement
Category: Iliad-Core-XHTMLElements

13.102.1 Iliad.ILTableRowElement: accessing

tag Answer ''tr".

13.103 Iliad.ILTextAreaElement

Defined in namespace Iliad
Superclass: Iliad.ILFormElementElement
Category: Iliad-Core-XHTMLElements

13.103.1 Iliad.ILTextAreaElement: accessing

tag Answer ‘’textarea”.

13.103.2 Iliad.ILTextAreaElement: accessing attributes

cols: anInteger
Not commented.

rows: anInteger
Not commented.

tabindex: anInteger
Not commented.

13.104 Iliad.ILTextElement

Defined in namespace Iliad
Superclass: Iliad.ILElement
Category: Iliad-Core-Elements

13.104.1 Iliad.ILTextElement: accessing

contents
Not commented.

contents: aString
Not commented.

tag Answer ‘tag’.

tag: aString
Not commented.

13.104.2 Iliad.ILTextElement: adding

add: anElement
Not commented.

text: aString
Not commented.

13.104.3 Iliad.ILTextElement: printing

printHtmlOn: aStream
Not commented.

13.105 Iliad.ILTitleElement

Defined in namespace Iliad
Superclass: Iliad.ILHTMLBuilderElement
Category: Iliad-Core-XHTMLElements

13.105.1 Iliad.ILTitleElement: accessing

tag Answer ''title".

13.105.2 Iliad.ILTitleElement: accessing attributes-events

onEvent: anEvent add: aString
Not commented.

13.106 Iliad.ILUrl

Defined in namespace Iliad
Superclass: Object
Category: Iliad-Core-lib-HTTP

I represent all portions of an URL as described by the RFC 1738. I include scheme, username, password, hostname, port, path, parameters, and fragment.

Portions of this code are based on code of Kazuki Yasumatsu and Paolo Bonzini.

Instance Variables scheme: <String> or nil username: <String> or nil password: <String> or nil hostname: <String> or nil port: <Integer> or nil path: <OrderedCollection> or nil parameters: <Grease.GRSmallDictionary> fragment: <String> or nil

13.106.1 Iliad.ILUrl class: decoding

decodePercent: aString
percent decodes the given String

13.106.2 Iliad.ILUrl class: instance creation

new Not commented.

13.106.3 Iliad.ILUrl class: parsing

absolute: aString
Parse aString as an absolute URL.

absolute: aString relative: aRelativeString
Take absolute URL aString and combine it with a relative path aRelativeString.

13.106.4 Iliad.ILUrl: accessing

fragment
Answer the fragment part of the URL.

fragment: aString
Not commented.

hostname
Answer the host part of the URL.

hostname: aString
Not commented.

parameters
Not commented.

password
Answer the password part of the URL.

password: aString
Not commented.

path Answer the path part of the URL.

path: aCollection
Set the path part of the URL to aCollection.

port Answer the port number part of the URL.

port: aNumber
Not commented.

scheme Answer the URL's scheme.

scheme: aString
we really expect a String here, Old versions (2.7) expected a Symbol you can still pass a Symbol and get away with it but don't expect this behavior to be supported in future versions. #greaseString is the 'correct' way to convert a Symbol to a String since #displayString will add a hash on VW

username
Answer the username part of the URL.

username: aString
Not commented.

13.106.5 Iliad.ILUrl: adding

addAllToPath: aCollectionOfStrings
convenice method to add a collection of strings to the path

addParameter: aString
Not commented.

addParameter: keyString value: valueString
Not commented.

addToPath: aString
Not commented.

13.106.6 Iliad.ILUrl: basic

, aString
Not commented.

13.106.7 Iliad.ILUrl: comparing

= anUrl
Not commented.

hash Not commented.

13.106.8 Iliad.ILUrl: converting

asString
Not commented.

greaseString
Not commented.

pathString
Answer the path converted to a string.

relativeTo: anUrl
Answer a path element collection relative from the receiver to anUrl.

13.106.9 Iliad.ILUrl: copying

postCopy
Not commented.

with: pathString
Not commented.

withParameter: aString
Not commented.

withParameter: aString value: valueString
Not commented.

withoutParameters
Not commented.

13.106.10 Iliad.ILUrl: encoding

printFragmentOn: aStream encoded: aBoolean
Not commented.

printOn: aStream encoded: aBoolean usingHtmlEntities: anotherBoolean
Not commented.

printParametersOn: aStream encoded: aBoolean usingHtmlEntities: anotherBoolean
Not commented.

printPathOn: aStream encoded: aBoolean
Not commented.

printServerOn: aStream encoded: aBoolean
Not commented.

13.106.11 Iliad.ILUrl: initialization

initialize
Not commented.

initializeFromString: aString
Not commented.

13.106.12 Iliad.ILUrl: parsing

parseFragment: aString
Not commented.

parseHost: aString
Not commented.

parseParameters: aString
Not commented.

parsePath: aString
Not commented.

13.106.13 Iliad.ILUrl: printing

printEncodedOn: aStream
Not commented.

printOn: aStream
Not commented.

printOn: aStream encoded: aBoolean
Not commented.

13.106.14 Iliad.ILUrl: removing

removeParameter: aString
Not commented.

13.107 Iliad.ILUrlBuilder

Defined in namespace Iliad
Superclass: Iliad.ILObject
Category: Iliad-Core-Sessions

13.107.1 Iliad.ILUrlBuilder class: accessing

actionKey
Not commented.

addRewriteRule: aBlock
Not commented.

ajaxUploadKey
Not commented.

hashKey
Not commented.

rewriteRules
Rewrite rules are used to replace patterns in the baseUrl

sessionKey
Not commented.

stateKey
Not commented.

13.107.2 Iliad.ILUrlBuilder class: defaults

defaultActionKey
Answer ‘’_action”.

defaultAjaxUploadKey
Answer ‘’_ajax_upload”.

defaultHashKey
Answer ‘’_hash”.

defaultSessionKey
Answer ‘’_session”.

defaultStateKey
Answer ‘’_state”.

13.107.3 Iliad.ILUrlBuilder: accessing

actionKey
Not commented.

hashKey
Not commented.

rewriteRules
Not commented.

sessionKey
Not commented.

stateKey
Not commented.

13.107.4 Iliad.ILUrlBuilder: generating

addDefaultParametersTo: anUrl
Not commented.

applyRewriteRulesTo: anUrl
Not commented.

baseUrl Not commented.

urlFor: aString

Not commented.

urlForAction: anAction

Not commented.

urlForAction: anAction hash: aHashString

Not commented.

urlForActionKey: aKeyString hash: aHashString

Not commented.

urlForRedirection: aString

Not commented.

13.107.5 Iliad.ILUrlBuilder: testing

shouldUseSessionField

Not commented.

13.108 Iliad.ILWidget

Defined in namespace Iliad
Superclass: Iliad.ILBuildable
Category: Iliad-Core-Buildables

I am a stateful graphical component.

"""""""""""""""""""""""""""""" " Building HTML with widgets " """"""""""""""""""""""""""""""

To build HTML override the #contents method, which should always return a view block, ie, a block which takes an element as parameter.

Example:

contents ^[:e || div | div := e div class: 'foo'. div h1: 'Bar'. div a text: 'do something'; action: [self doSomething]]

See Iliad.ILElement hierarchy (Especially Iliad.ILHTMLBuilderElement) for more information about building HTML with elements.

#contents method should *never* be called from the outside. Use #build instead. For instance, to build a sub-widget in a view block, you should write something like:

contents [^[:e | e build: mySubWidget]]

"""""""""""""""" " Control flow " """"""""""""""""

I can show (display instead of me) other widgets with #show* methods or answer to widgets that called me with #answer.

When using the javascript layer, call #markDirty whenever my state change, so I will be updated on AJAX requests.

Widgets which states depend on me can be automatically rebuilt whenever I am marked as dirty (see #addDependentWidget:).

""""""""""""""" " Decorators " """""""""""""""

I can also have decorators that may modify my behavior. A decorator can be added to the decoration chain with #decorateWith:.

13.108.1 Iliad.ILWidget: accessing

dependentWidgets
Not commented.

id Not commented.

id: aString
Not commented.

owner Answer the widget which shows me. if any

owner: aWidget
Not commented.

state Not commented.

stateRegistry
Not commented.

widget Answer the receiver.

widgetFor: aBuildable
Convenience method. This is useful for building anonymous widgets. ex: myWidget := self widgetFor: [:e | e h1: 'Hello world!']

13.108.2 Iliad.ILWidget: building

buildContents
Do *not* override this method. Use #contents instead

buildHead: aHead
Not commented.

contents
Override this method to add contents to your widget

fullContents
Do *not* override this method. Use #contents instead

scripts
Answer a collection of strings. Override in subclasses to add scripts to load with the widget

styles
Answer a collection of strings. Override in subclasses to add styles to load with the widget

13.108.3 Iliad.ILWidget: control flow

addDependentWidget: aWidget
Add <aWidget> to my dependent widgets. Each dependent widget will be rebuilt on AJAX requests whenever I am rebuilt

answer
Give the control back to the owner, i.e, the widget which showed the receiver. Answer self

answer: anAnswer
Give the control back to the owner, i.e, the widget which showed the receiver. Answer <anAnswer>

append: aWidget
Insert <aWidget> after the receiver

append: aWidget onAnswer: aBlock
Insert <aWidget> after the receiver

confirm: aString ifTrue: aBlock
Not commented.

confirm: aString ifTrue: aBlock ifFalse: anotherBlock
Not commented.

handleAnswer: anAnswer
Answer 'nil'.

inform: aString
Not commented.

prepend: aWidget
Insert <aWidget> before the receiver

prepend: aWidget onAnswer: aBlock
Insert <aWidget> before the receiver

removeDependentWidget: aWidget

Not commented.

retrieveControl

Give the control back to the receiver, and make any showed widget answer nil

show: aWidget

Show another widget instead of the receiver. The receiver is also implicitely marked dirty

show: aWidget onAnswer: aBlock

Show another widget instead of the receiver and catch the answer in <aBlock>. The receiver is also implicitely marked dirty

show: aWidget onAnswer: aBlock delegator: aDelegator

Not commented.

13.108.4 Iliad.ILWidget: copying

postCopy

Not commented.

13.108.5 Iliad.ILWidget: decorators

decorateWith: aDecorator

Not commented.

decorateWith: aDecorator during: aBlock

Not commented.

decoratorsDo: aBlock

Not commented.

removeDecorator: aDecorator

Remove <aDecorator> from the decoration chain, except if <aDecorator> is the initial one

withDecoratorsDo: aBlock

Not commented.

13.108.6 Iliad.ILWidget: defaults

rootElementClass

Answer ‘ILDivElement’.

13.108.7 Iliad.ILWidget: initialization

initialize

Not commented.

13.108.8 Iliad.ILWidget: printing

printJsonOn: aStream

Not commented.

13.108.9 Iliad.ILWidget: states

markDirty

Mark the receiver as ’dirty’, so the widget will be rebuilt on Ajax requests. You do not need to mark subwidgets as dirty, they will be rebuilt together with the receiver

registerState

Not commented.

13.109 Iliad.ILXmlElement

Defined in namespace Iliad
Superclass: Iliad.ILElement
Category: Iliad-Core-Elements

13.109.1 Iliad.ILXmlElement: accessing

contentType
Not commented.

tag Answer ‘tag’.

tag: aString
Not commented.

xmlTag Not commented.

13.109.2 Iliad.ILXmlElement: converting

respondOn: aResponse
Not commented.

Index