

CAPPASITY iOS FRAMEWORK (Swift)

v0.3

Contents

CAPPASITY iOS FRAMEWORK (Swift).....	1
License	1
Overview	2
Importing framework	3
The CappasityDelegate Protocol	3
protocol CappasityDelegate	3
User Account Login.....	3
protocol CappasityServiceDelegate	3
class CappasityService	3
Obtaining a List of User Models	4
protocol CappasityModelsListDelegate	4
class CappasityModelsList	4
Working with the Model	5
protocol CappasityModelDelegate.....	5
class CappasityModel	5
Displaying the Model.....	8
protocol CappasityModelViewDelegate.....	8
class CappasityModelView	8
Error List	10
Obtaining the Token	11

License

Copyright (c) 2017 - 2018, Cappasity Inc. All rights reserved.

Redistribution and use in binary form, without modification, are permitted provided that the following conditions are met:

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Cappasity Inc., nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Overview

CappasityFramework contains functionality for working with models on the Cappasity platform, i.e.:

- obtaining the models,
- obtaining the properties of a given model, editing the model, deleting the model,
- displaying the model.

Logging into your user account is done by providing the CappasityService class with the generated token, by sending it to the login(with) method. The login is effective until the application is closed.

The entire set of user models may be obtained by placing consecutive requests for the subsequent n models, where n should be no greater than 100. Functionality for obtaining a list of models is contained in the CappasityModelsList class.

A specific model may be obtained via its id or sku. To do this, you need to make an instance of the CappasityModel class, using the corresponding initializers init(withId:delegate) or init(withSku:delegate) and calling the receiveInfo() method.

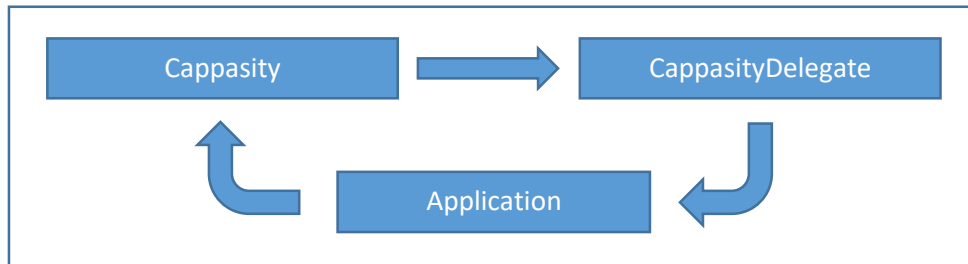
CappasityModel allows access to the following model information:

- sku,
- model title,
- model access type,
- model embed code.

The model's sku, title, and access type may be modified via the update(title:sku:access) method. The remove() method deletes the model.

To display the model, you may use the embed code and incorporate it into your application yourself. Alternatively, you can use the CappasityModelView class. The CappasityModelView class inherits the UIView class, so it can be added into the application either through the storyboard or directly in the program.

Interaction with the classes is implemented through delegates.



Importing framework

```
import Cappasity
```

The CappasityDelegate Protocol

protocol CappasityDelegate

A protocol of shared methods.

All delegates for the main classes are inherited from the CappasityDelegate protocol.

Methods

`cappasityError(code: Int, description: String)`

Tells the delegate of an error occurring during program execution.

Parameters:

- code – error number
- description – error description

`cappasityMessage(_ message: String)`

Tells the delegate certain information regarding CappasityService state

User Account Login

protocol CappasityServiceDelegate

A collection of methods that may be called by the CappasityService class object.

Overview

The `loginCompleted()` method is called in the case of a successful account login.

Inherits the CappasityDelegate protocol.

Methods

`loginCompleted ()`

Tells the delegate of the successful account login.

class CappasityService

Performs a token-based login to the Cappasity system.

Overview

The login(with) method is used to log into the user account. In the case of a successful login, the loginCompleted() method of the CappasityServiceDelegate delegate is called, which instance is given during the initialization of the object.

Methods

init(delegate: CappasityServiceDelegate?)

Returns an instance of the CappasityService class.

Parameters:

- delegate – the delegate object.

login(with token: String)

Sends a request for a token-based login to the user account.

Parameters:

- token – the user token.

Obtaining a List of User Models

protocol CappasityModelsListDelegate

A collection of methods that may be called by the CappasityModelsList class object.

Overview

The method received(models) is called in the case of a successful request.

Inherits the CappasityDelegate protocol.

Methods

received(models: [CappasityModel])

Returns the models obtained.

Parameters:

- models – the array of models

class CappasityModelsList

Obtaining of the user models.

Overview

The registered user may obtain a list of all owned models sorted by the order in which the models were created.

Properties and methods

init?(delegate: CappasityModelsListDelegate)

Returns an instance of the CappasityModelsList class in case of a successful account login. Returns “nil” otherwise.

Parameters:

- delegate – the delegate object.

nextModels(limit: Int, offset: Int? = nil)

Request to obtain the next set of models.

Parameters:

- limit – the number of models requested. Range of values: [1, 100].
- offset – the sequential rank of the model from which the next set commences. In the case that this value is equal to “nil”, the models requested are those that are next in line after the last previously obtained model. Range of values: [0,...)

isValid() -> Bool

Verification of class validity. The method returns “false” in the case of an error during the request, and returns “true” otherwise.

clearStatus()

A reset of all object properties to their original state. Following this operation, the nextModels method will return the models from the very beginning of the list in the case that the “offset” parameter is not specified.

Working with the Model

protocol CappacityModelDelegate

A collection of methods that may be called by the CappacityModel class object.

Overview

Inherits the CappacityDelegate protocol.

Methods

modelInfoReceived()

Tells the delegate of having successfully obtained the requested model information.

modelUpdated()

Tells the delegate of having successfully updated the model properties.

modelRemoved()

Tells the delegate of having successfully deleted the model.

previewReceived(preview: UIImage, userInfo: AnyObject?)

Returns the requested model preview.

Parameters:

- preview – model preview image
- userInfo – user information entered during the preview request.

class CappacityModel

Working with the model.

Overview

The object may be initialized via the model's id or sku. The receiveInfo() method should then be called to obtain the model information.

```
let model = CappacityModel(id: "123456789", delegate: self)
```

```

model.receiveInfo()
...
//the object is ready for use if the request has been successfull (the delegate received the
//modelInfoReceived() method).

print(model.sku)

```

There is no need to call the receiveInfo() method if the model has been obtained via a request for a list of user models.

```

//MARK: -- CappasityModelsListDelegate method
func received(models: [CappasityModel]) {
    if models.isEmpty == false {
        print(models[0].sku)
    }
}

```

Since the delegate has not been set during the initialization of the object in this case, this should be done directly if needed.

```

let model = models[0]
model.delegate = self // weak reference

```

Properties and methods

var delegate: CappasityModelDelegate?
The delegate object.

id: String?
Model id.

title: String?
Model title.

sku: String?
Model sku.

access: FileAccessType?
Model visibility setting.
FileAccessType values:

publicAccess	model is visible to everyone
unlistedAccess	model is accessible only through a direct link
privateAccess	model is visible only to owner

init(withId id: String, delegate: CappasityModelDelegate?)
Returns an instance of the CappasityModel class
Parameters:
- id – model id.
- delegate – the delegate object.

init(withSku sku: String, delegate: CappasityModelDelegate?)
Returns an instance of the CappasityModel class
Parameters:
- sku – model sku.

- delegate – the delegate object.

receiveInfo()

Sends a request to obtain the model information. Calls the delegate method modelInfoReceived() in the case of a successful execution.

update(title: String? = nil, sku: String? = nil, access: FileAccessType? = nil)

Sends a request to update the model information. Calls the delegate method modelUpdated() in the case of a successful execution.

Parameters:

- title – new model title
- sku – new model sku
- access – new model access type

remove()

Sends a request to delete the model. Calls the delegate method modelRemoved() in the case of a successful execution.

getPreview(widthPx: UInt, heightPx: UInt, userInfo: AnyObject? = nil)

Sends a request to obtain the model preview. Calls the delegate method previewReceived(preview:userInfo) in the case of a successful execution.

Parameters:

- widthPx – width of requested preview in pixels.
- heightPx – height of requested preview in pixels.
- userInfo – user information, returned together with the obtained model preview.

getPreview(widthPt: UInt, heightPt: UInt, userInfo: AnyObject? = nil)

Sends a request to obtain the model preview. Calls the delegate method previewReceived(preview:userInfo) in the case of a successful execution.

Parameters:

- widthPt – width of requested preview in points.
- heightPt – height of requested preview in points.
- userInfo – user information, returned together with the obtained model preview.

embedCode(with width: String? = nil,

height: String? = nil,

autoRotate: Bool? = nil,

autoRun: Bool? = nil,

closeButton: Bool? = nil,

showLogo: Bool? = nil,

autoRotateTime: Float? = nil,

autoRotateDelay: Float? = nil,

autoRotateDir: Int? = nil,

hideFullScreen: Bool? = nil,

hideAutoRotateOptions: Bool? = nil,

hideSettingsButton: Bool? = nil,

enableImageZoom: Bool? = nil,

zoomQuality: Int? = nil,

hideZoomOption: Bool? = nil) -> String?

Returns the embed code of the model with the specified parameters.

Parameters:

- width – window width, given in pixels or percentages. For example, “600px” or “100%”.
- height – window height, given in pixels or percentages.
- autoRotate – switch the automatic rotation of the object on/off.
- autoRun – switch autorun on/off.
- closeButton – show/hide the Close button.
- showLogo – show/hide the logo.
- autoRotateTime – time for a single rotation of the object during automatic rotation. Value specified in seconds. Available when autoRotate is enabled.
- autoRotateDelay – time delay prior to the start of the automatic rotation. Value specified in seconds. Available when autoRotate is enabled.
- autoRotateDir – direction of the automatic rotation. 1 – clockwise, -1 – counterclockwise. Available when autoRotate is enabled.
- hideFullScreen – show/hide the Full Screen button.
- hideAutoRotateOptions – show/hide the Automatic Rotation button. Available when autoRotate is enabled.
- hideSettingsButton – show/hide the Settings button.
- enableImageZoom – switch zoom mode on/off.
- zoomQuality – zoom quality. Possible values:
1 – SD quality,
2 – HD quality.
- hideZoomOption – show/hide zoom options.

Displaying the Model

protocol CappasityModelViewDelegate

Delegate, which methods can be called by the CappasityModelView class object.

Overview

Inherits the CappasityDelegate protocol.

class CappasityModelView

Displays the model with the specified parameters in a web browser.

Overview

The CappasityModelView class inherits UIView. It can be appointed to an object directly in the storyboard. The “set” method is used to specify the model to be displayed, as well as the display parameters.

In order to enable transition to Cappasity's official website when clicking on the logo, viewController property must be set.

Properties and methods

var delegate: CappasityModelViewDelegate?

The delegate object.

var viewController: UIViewController?

The UIViewController object. Must be set for working logo button.

Clicking on the Cappasity logo will have no effect if this object is not specified.

func stopLoading()

Stops the loading of the model.

set(_ model: CappasityModel,
 autoRotate: Bool? = nil,
 autoRun: Bool? = nil,
 showLogo: Bool? = nil,
 autoRotateTime: Float? = nil,
 autoRotateDelay: Float? = nil,
 autoRotateDir: Int? = nil,
 hideAutoRotateOptions: Bool? = nil,
 hideSettingsButton: Bool? = nil,
 enableImageZoom: Bool? = nil,
 zoomQuality: Int? = nil,
 hideZoomOption: Bool? = nil)

Specifies the model to be displayed and the display parameters.

Parameters:

- model – model to be viewed.
- autoRotate – switch the automatic rotation of the object on/off.
- autoRun – switch autorun on/off.
- showLogo – show/hide the logo.
- autoRotateTime – time for a single rotation of the object during automatic rotation. Value specified in seconds. Available when autoRotate is enabled.
- autoRotateDelay – time delay prior to the start of the automatic rotation. Value specified in seconds. Available when autoRotate is enabled.
- autoRotateDir – direction of the automatic rotation. 1 – clockwise, -1 – counterclockwise. Available when autoRotate is enabled.
- hideAutoRotateOptions – show/hide the Automatic Rotation button. Available when autoRotate is enabled.
- hideSettingsButton – show/hide the Settings button.

- enableImageZoom – switch zoom mode on/off.
- zoomQuality – zoom quality. Possible values:
 - 1 – SD quality,
 - 2 – HD quality.
- hideZoomOption – show/hide zoom options.

Error List

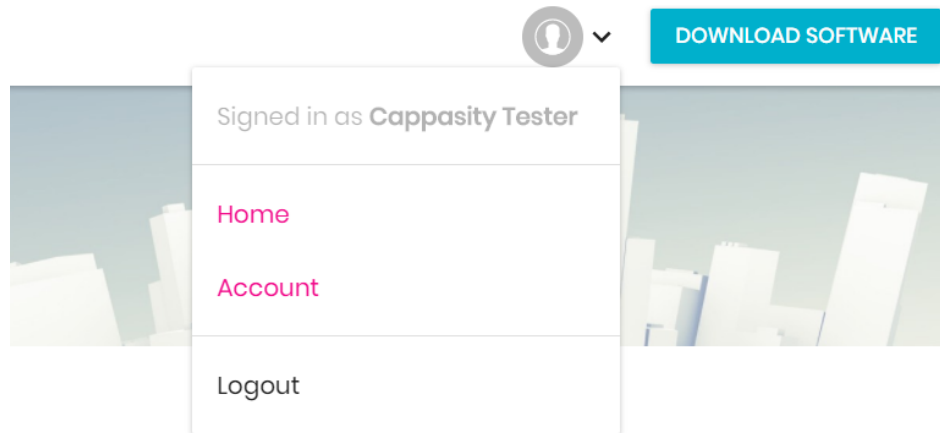
Codes of potential errors encountered during program operation.

Code	Description
102	The login was unsuccessful.
103	You must log into the account first.
104	The token should not be empty.
105	The model has not been initialized. Call receiveInfo to obtain the model parameters.
106	The sku should not be empty.
107	The account name should not be empty.
108	The account name should only have 0-9, a-z/A-Z, and '.' characters.
109	The sku should only have 0-9, a-z/A-Z, '.', '_', and '-' characters.
110	The account name should contain at least 3 characters.
111	The password should contain at least 6 characters.
112	The title should not be empty or contain more than 75 characters.
113	The sku should contain no more than 50 characters.
150	The request was unsuccessful.
151	The request to obtain the model was unsuccessful.
152	The request to obtain the list of models was unsuccessful.
153	The request to obtain the model preview was unsuccessful.
154	The request to obtain the model parameters was unsuccessful.
155	The deletion of the model was unsuccessful.
156	The model parameter update was unsuccessful.
157	The modification of the model access type was unsuccessful.
158	The request to obtain the model's embed code was unsuccessful.
159	The video could not be processed.
160	Uploading model to the cloud server was unsuccessful.
161	The model's embed code could not be obtained.
162	The user search request was unsuccessful.
163	The file is being processed. Please wait for up to 15 minutes.
164	You can modify the file access settings if your activated paid account is Basic or higher.
300	Not enough memory.
301	Your request cannot be processed.

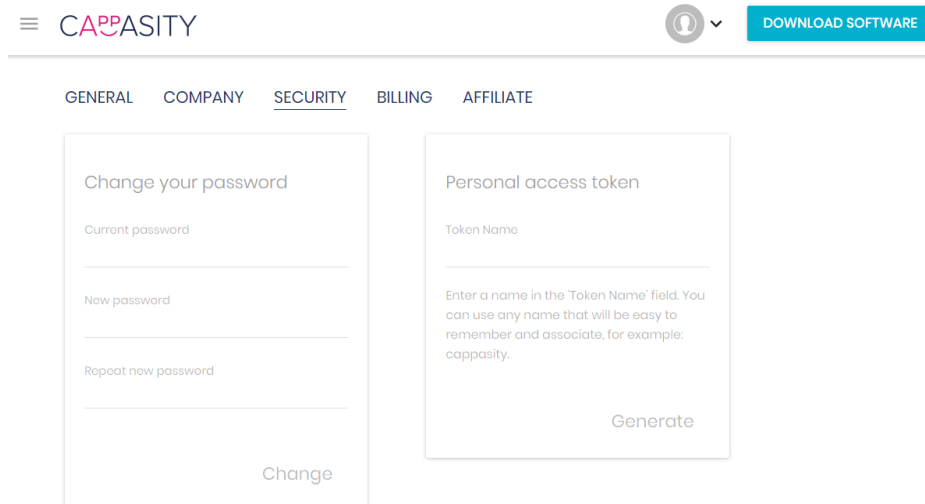
Obtaining the Token

To generate the token, you need to perform the following steps:

- 1) sign up for an account on the Cappasity platform at <https://3d.cappasity.com/>, or log into your existing account,
- 2) click on the Account tab in the menu,



- 3) click on the Security tab,



- 4) enter the token name and click on the Generate button,
- 5) copy the token obtained.