

TEACH  N MARS

CUSTOM WEB HOME SCREEN

TEACH ON MARS - MOBILE LEARNING APPS

© 2018 Teach on Mars

Integration Guide	3
Introduction	4
Why a custom home screen?	4
Why a "web" home screen?	4
Design guidelines	5
Think mobile first!	5
Consistency	5
Test	5
Interact with the application	6
How to call these methods	6
Catalog	6
ToM.home.getCategories()	6
ToM.home.displayCategory(categoryId)	7
ToM.home.getTrainingCourses()	9
ToM.home.getTrainingCoursesByCategory(categoryId)	10
ToM.home.displayTrainingCourse(trainingCourseId)	10
Communications	12
ToM.home.getFeaturedCommunications()	12
ToM.home.displayCommunication(communicationId)	13
Profile	13
ToM.home.getProfile()	13
ToM.home.displayProfile()	14
Wall	15
ToM.home.displayWall()	15
Shake'n'Learn	16
ToM.home.isShakeAndLearnPossible()	16
ToM.home.shakeAndLearn()	16
Other methods	17
ToM.env.getAll()	17
ToM.env.get(variable)	17

Integration Guide

Design guidelines

Interact with the Teach on Mars app

Introduction

The goal of this document is to provide a complete guide to the integration of a custom home screen for a Teach on Mars mobile or web application. It is intended for Teach on Mars clients and web agencies hired to create an original home interface.

Why a custom home screen?

The Teach on Mars application is great out of the box, and its home screen may already be configured to suit your needs and graphical charter. But we thought some of you might like something more.

So here comes the *Custom web home* that allows you to create pretty much any kind of home screen as long as it is web based. This web home will be able to interact with the standard part of the application.

Why a "web" home screen?

HTML5, CSS and Javascript are both very common and versatile technologies. HTML5 and CSS allows you to be very creative and build a unique user experience that includes your brand graphical identity. You will find many agencies and freelance developers that will gladly create your custom web home screen to enhance your learners' experience.

Design guidelines

Here are some quick tips to make sure the user experience of your home screen is perfect.

Think mobile first!

Your home screen will be displayed on a large variety of devices and screen sizes. So you need to create a responsive design that will work on different device dimensions and width/height ratios.

Questions that might help you design your home screen:

- Do you know what kind of device your users will use?
- Is it only tablets? Phones? Both?
- Are these standard devices? Does each user bring his own device?

The *Mobile first* rule applies to the graphic design, but also to the UX design.

Mobile first also means offline

Internet connectivity is a real issue with mobile devices, so designing mobile first also means designing something that can work offline. So, all files required for your custom home screen must be embedded in the application.

Mobile first but not only

If your design is so mobile first that it becomes hard to adapt it to a desktop sized web app, you might consider creating two custom home screen designs: one optimized for mobile devices and the other adapted to desktop sized screens.

Consistency

Even though the first impression of the user is very important, let's not forget the whole user experience. The home screen is the entry point to the application but behind it, the other screen that the user will browser through are standard Teach on Mars design. So, it is important to keep consistency between the home screen and the rest of the application.

Think about the font family you are using, the style and size of the interface labels, the style of the buttons, etc. Also keep consistency between the color palette you are using and the one that has been configured in the application.

Test

It might be obvious to say this but it is essential to test the display of your custom home on as many device as possible. And not only the display, but also the action of the links or buttons that you've put in your custom home screen.

You don't have to test on every single device, but try to do it on a representative collection of devices (device platform and screen sizes). If you know what kind of device the target population uses, it is obvious what your tests should focus on. If you don't, there are plenty of statistics showing what are the most commonly used devices, so you should start with that.

Interact with the application

It is possible to use a JS library to fetch useful data from the Teach on Mars application. This way, your custom home can become dynamic and keep up with the updates that are made from the Mission Center.

In this section, we will cover how to use the ToM Home API to integrate data from the Mission Center in you custom home screen.

So here we go.

How to call these methods

The ToM Javascript library is organized in namespaces that each have a purpose.

First things first, there is some basic principles to know about the **ToM.home** library namespace. All its methods actually return a *Promise*, so they should be called this way

```
ToM.home.getCategories()
  .then(function(categories) {
    _.each(categories, function(category) {
      displayCategoryToHome(category);
    })
  })
```

This example uses lodash methods, but it's up to the developer to choose what's best.

In the webapp

The custom web home in the web app is displayed in an iframe, so before you can start calling the following methods in the webapp, you need to get a link to the library object called *ToM* in your environment. To do that, you need to reference the *ToM* object in the parent window like this:

```
window.addEventListener('load', function () {
  window.ToM = window.ToM || (window.top && window.top.ToM) || {};
});
```

Catalog

ToM.home.getCategories()

Returns a flat collection of categories that the learner have access to. Categories describe the tree-like structure of the catalog and contain all the training courses.

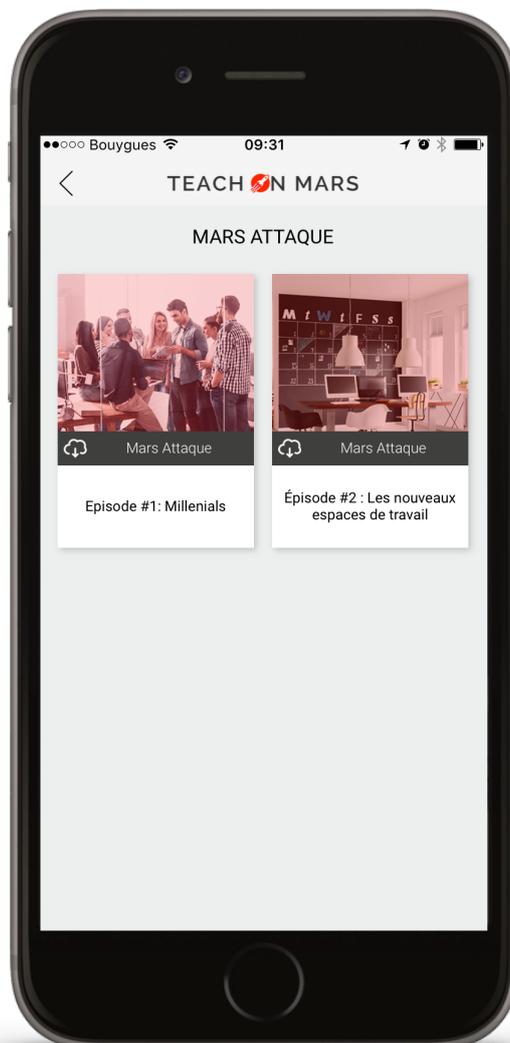
The image of the category is available at the given URL but, it requires the Web Services host prefix to be complete:

<https://my-userdata.teachonmars.com/api/v3/category/image/cover/984265767659ae56>

```
[
  {
    "id": "categoryId",
    "title": "categoryTitle",
    "parent": "parentCategoryId" or null if root category
    "children": [childCategoryObject1,
childCategoryObject2, ...],
    "image": "category/image/cover/984265767659ae56"
  },
  ...
]
```

ToM.home.displayCategory(categoryId)

Navigates to the defined category. This allows you to create a link from your custom home to a category screen, showing the training courses in this category.



ToM.home.getTrainingCourses()

Returns a collection of training courses.

```
[
  {
    "id": "my-training-course",
    "title": "My Training Course",
    "availableLanguagesCodes": [
      "en"
    ],
    "badgesEnabled": true,
    "badges": [
      {
        "id": "training-complete-my-training-course",
        "name": {
          "en": "Training completed!"
        },
        "description": {
          "en": "The training course 'My Training Course' has
been completed."
        },
        "type": "training-complete",
        "picto": "ui/badges/badge-training-complete.png",
        "pictobig": "ui/badges/badge-training-complete-
big.png",
        "conditions": {
          "training": "my-training-course"
        },
        "timestamp": 1520257579
      }
    ],
    "coverImage": "training/my-training-course/image/cover/
b38b16f4e478455a1d5456873581963aa285bf547ff71d2f3ed5a42d9acd4289",
    "defaultLanguageCode": "en",
    "description": "Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Quisque sed pellentesque tellus. Donec condimentum
eget mi id venenatis. Integer convallis ex quis felis hendrerit
dapibus. Phasellus turpis nisi, pretium id vehicula eget, porttitor
a arcu. Phasellus odio elit, dictum sit amet accumsan in, maximus
sed augue.",
    "descriptionImage": null,
    "isLive": false,
    "order": 0,
    "rankingEnabled": false,
    "sandbox": false,
    "thumbnailImage": "training/my-training-course/image/thumbnail/
7f01eb819246ea218ed7e6f37631d69b0a643cf9d0e5aba5b3b81a8a31b113c3",
    "timestamp": 1517915183,
    "type": "coaching",
    "version": 1
  }
]
```

The images of the training course is available at the given URL but, it requires the Web Services host prefix to be complete. Here's an example with the thumbnail image:

https://my-userdata.teachonmars.com/api/v3/training/my-training-course/image/thumbnail/7f01eb819246ea218ed7e6f37631d69b0a643cf9d0e5aba5b3b81a8a31b113c3

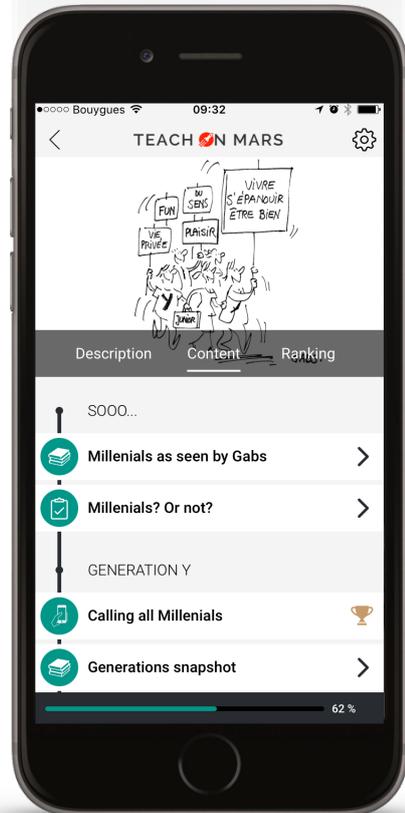
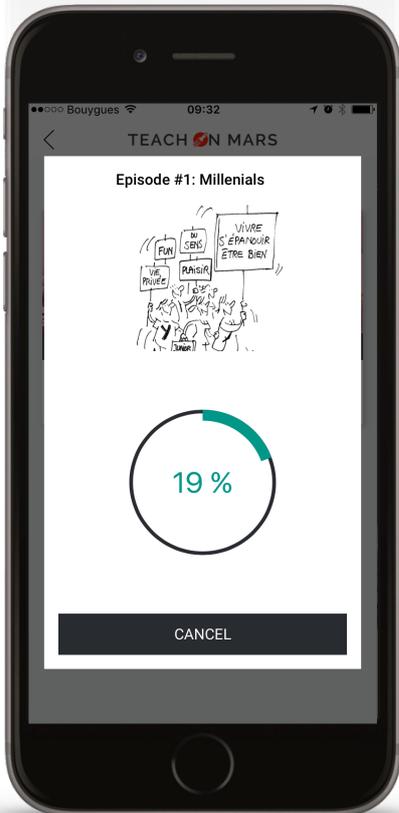
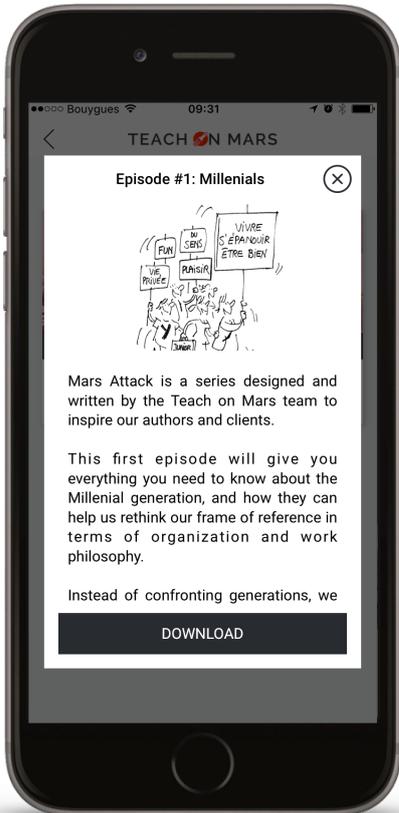
ToM.home.getTrainingCoursesByCategory(categoryId)

Returns a collection of training courses that are in the given category.
Same format as the *ToM.home.getTrainingCourses()* method.

ToM.home.displayTrainingCourse(trainingCourseId)

Navigates to the given training course screen. If the training course content needs to be downloaded, the application will automatically go through the download process before displaying the training course screen.

On the next page are some screenshots of the complete training course download process. Of course, if the user has already downloaded the training course, this method navigates straight to the last step.



Communications

ToM.home.getFeaturedCommunications()

Returns a collection of communications that have been tagged as *featured* in the Mission Center.

```
[
  {
    "avatar": "communication/avatar/longtext/
7a49e231f69fb51e07a5b11843bcdf8a",
    "data": {
      "imageSize": {
        "height": 480,
        "width": 768
      }
    },
    "id": "4714",
    "image": "communication/4714/image/
97bea2a61b610c6206f8e5e0ab93f363de5cb322d172f147ca22e04c18512a88",
    "link": "",
    "longText": "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Morbi viverra eleifend turpis eu tincidunt. Quisque varius
nisi elementum dolor posuere luctus. In quis vestibulum arcu, eu
tincidunt tellus. Sed at sem et neque suscipit volutpat.
Pellentesque in lectus id metus ultricies rutrum.",
    "shortText": "My communicaiton with a long text",
    "timestamp": 1514395241,
    "trainingId": "",
    "type": "longtext"
  },
  {
    "avatar": "communication/avatar/training/
7a49e231f69fb51e07a5b11843bcdf8a",
    "data": {
      "imageSize": {
        "height": 480,
        "width": 768
      }
    },
    "id": "4404",
    "image": "communication/4404/image/
c5aae3c8f61b28cdd4191ed1708da5be8e79d8dad8d1884bbcc98b6bbe657008",
    "link": "",
    "longText": "",
    "shortText": "\"Quiz Battle\": Try it now!",
    "timestamp": 1513614857,
    "trainingId": "quiz-battle",
    "type": "training"
  }
]
```

ToM.home.displayCommunication(communicationId)

Displays a given communication. What is displayed depends on the type of communication.

Type	Behavior
Article	Displays the article with its header image on a new screen
Training	Displays the training course screen just like ToM.home.displayTraining()
Web Link	Opens the link in a web view. The web view can be closed and refreshed

Profile

ToM.home.getProfile()

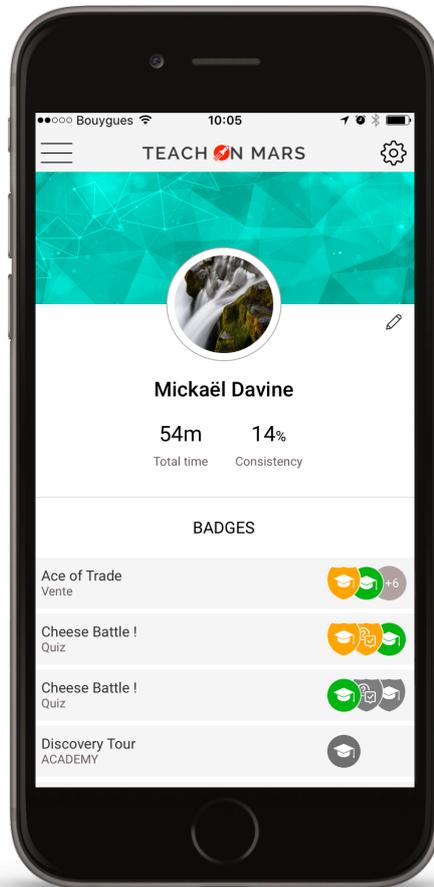
Returns information about the learner and some basic learning data.

The object *trainings* contains data for each training course that the learner has started working on.

```
{
  "learner": {
    "avatar": "device/learner/5205dda0-db62-11e7-a799-25541a29a5fd/
avatar/956872937255892",
    "email": "jon.doe@company.com",
    "firstname": "John",
    "lang": "en",
    "lastname": "Doe",
    "learnerId": "5205dda0-db62-11e7-a799-25541a29a5fd",
    "login": "john.doe",
    "metadata": {
      "company": "",
      "country": "US"
    },
    "nickname": "John"
  },
  "lastLaunchedTraining": "my-training-course",
  "points": 45,
  "progress": 67,
  "time": 349,
  "trainings": {
    "my-training-course": {
      "trainingId": "my-training-course",
      "badges": ["training-complete-my-training-course"],
      "points": 13,
      "progress": 95,
      "certified": false,
      "time": 132,
      "firstLaunched": 1514851200,
      "lastLaunched": 1514937600
    }
  }
}
```

ToM.home.displayProfile()

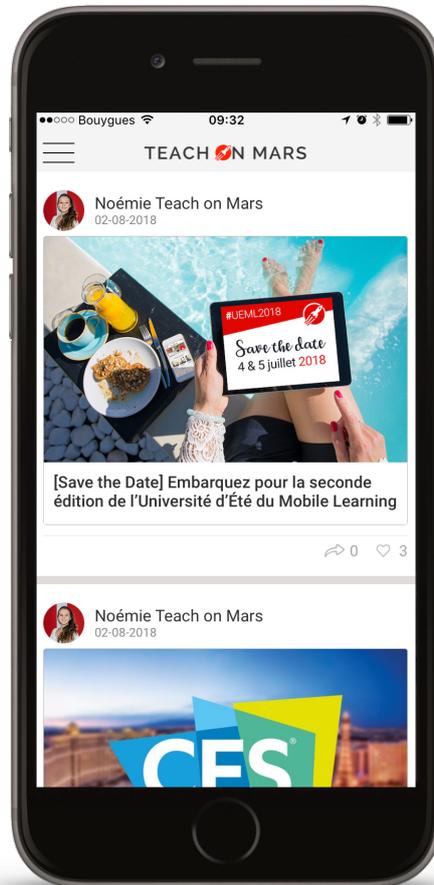
Navigates the application to the logged in learner profile screen.



Wall

ToM.home.displayWall()

Navigates the application to the Wall screen.



Shake'n'Learn

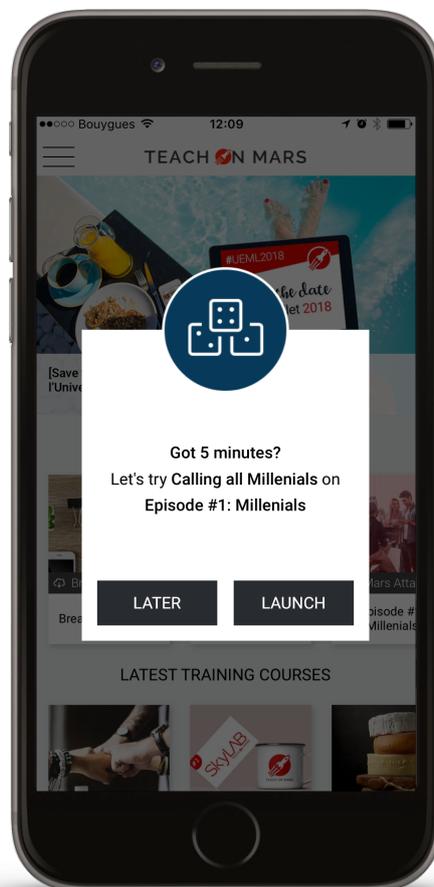
Shake'n'Learn is a feature of the Teach on Mars Enterprise solution that allow the user to quickly launch a scored activity that has already been attempted by shaking their mobile phone while on the home screen. It is a fun way to review a random lesson.

ToM.home.isShakeAndLearnPossible()

Return true or false whether an activity can be launched through the Shake'n'Learn feature.

ToM.home.shakeAndLearn()

Launches a random scored activity that have already been completed before. It is a way to go over some gamified content again. This replaces the shake gesture and launches the Shake'n'Learn action.



Other methods

ToM.env.getAll()

Returns all environment variables in a single object

Variable	Description
APPLICATION_SERVER_ID	Application server ID
APPLICATION_VERSION	Application version
EMAIL	Email for current logged in user
FIRST_NAME	First name for current logged in user
LANGUAGE	Language setted in the application
LAST_NAME	Last name for current logged in user
LEARNER_ID	The unique identifier for current logged in user
LOGIN	Login of current logged in user
USER_OS	OS on witch the application is running
metadata.CODE_META	Display the value of a metadata (CODE_META is case sensitive

ToM.env.get(variable)

Returns the value of the given variable.