

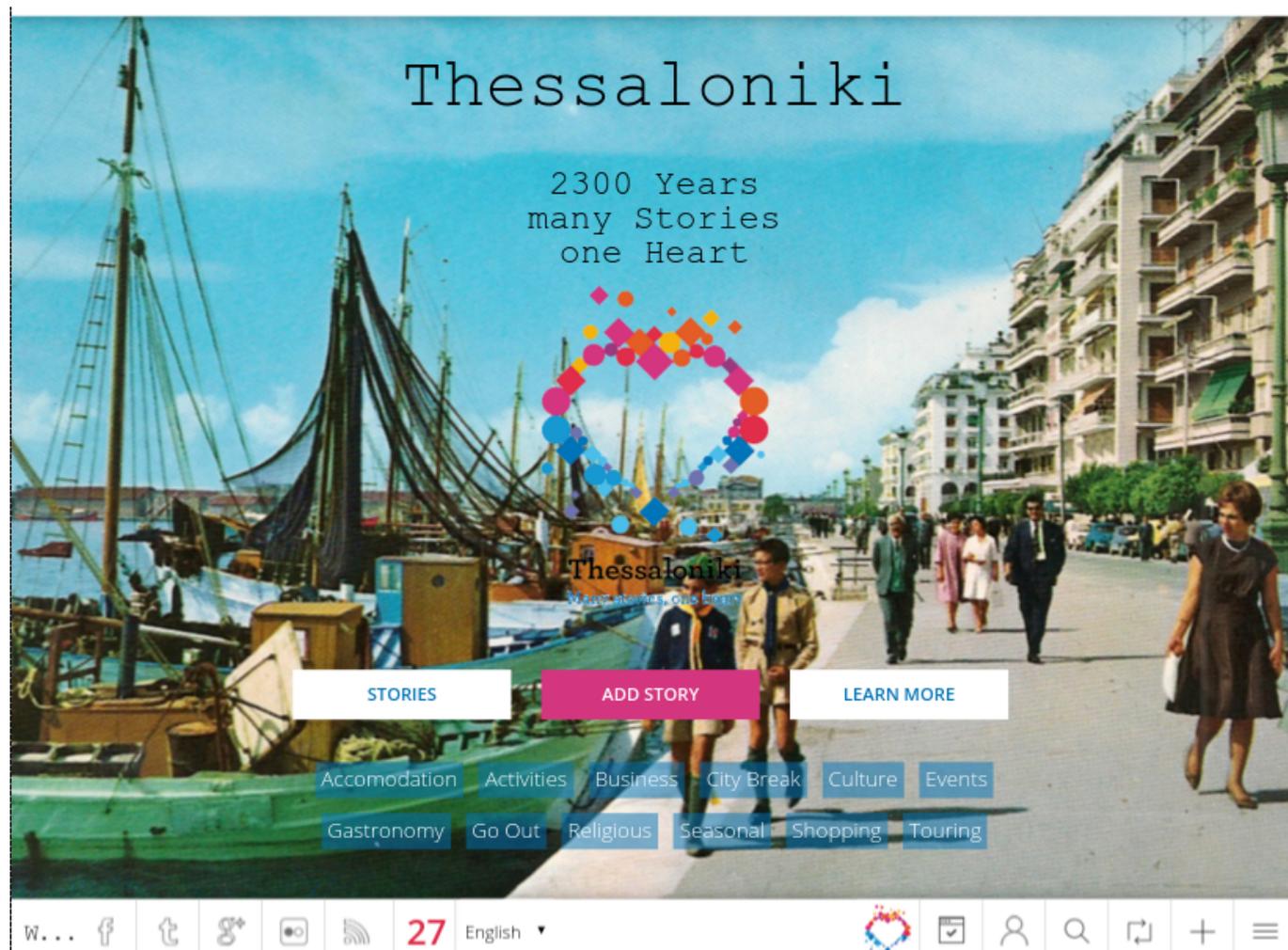
This is a copy of the online Documentation at manystories.readthedocs.org

ManyStories

- Online website: manystoriesoneheart.gr
- Documentation: manystories.readthedocs.org
- Github organization: github.com/manystories
- Apps (using the APIs): manystories.github.io/apps

This project is a try to give back to our city. It is an official submission for the 2016 [Apps4Thessaloniki competition - Tourism edition](#). We are not affiliated with any public sector organization of the city.

Logo and other media assets are owned by the [Municipality of Thessaloniki](#). Please ask the owners before use them.



About

Thessaloniki is a city full of stories that have been unfolding over 2300 years. They are the stories of her people, which we can today discover, taste and seek out in her historical monuments, strolling down her picturesque backstreets or chatting with passers-by.

Thessaloniki has many stories to tell us.

This is a city with a proud historical past written by the multiculturalism of her people. A swathe of societies –Ancient Greeks, Romans, Byzantines, Ottomans, Jews, Armenians and many others have bequeathed Thessaloniki a multi-coloured, multifarious, multidimensional culture.

This is a hospitable city, with warm people, an optimistic spirit and a carefree attitude to life – you come across this everywhere you go - walking on the beach, having a coffee, going down the city centre alleyways.

This is a port-city in the embrace of the Aegean, offering up many Mediterranean destinations. And more. Thessaloniki is bathed in the shadow of Mount Olympus – the mountain of the gods, is next to Chalkidiki – the best known resort in the Balkans, and is close to areas of rich natural scenery, such as Delta Axiou.

This is a center of ideas which concentrate themselves in the restless spirit of modern art – the Thessaloniki International Film Festival, but also a business center of innovation – the Thessaloniki International Trade Fair.

All these elements come together to make up the city's heart, as well as Thessaloniki's new logo.

A heart-mosaic, made up of pebbles – the stories of the city.

And her spirit sends us a message: Many stories, one heart.

Because this is Thessaloniki.

A heart full of stories.

A heart that beats strongly.

[Listen to it.](#)

What is this project?

Browsing this project online you may think this is another Tourism portal, a simple Social media app, a simple data hub or just a website.

You may not be wrong. Actually, it's all the above but many more also.

If you read the above "symbolism" about Thessaloniki as a Heart of Stories (heart of pebbles) then you already know that the important thing on the Heart is the Story.

Stories about the city, about her people, about the events that happened or are going to happen. Stories about love and hate, from good and bad times of the history.

Using technology and imagination we tried to transfer this symbolism to a real world web application.

Storytelling, internet, social media, crowdsourcing and linked data are some only of the tools we used to make fulfill this achievement.

We just love Thessaloniki and we love to spread this love to anyone else interested. We are not affiliated with any public sector organization of the city. We love to see the Stories of Thessaloniki on the first place of the history, where they belong.

We love to see opened our Stories to everyone and not keep them hidden on our minds or in some organizations locked archives.

Because there is no better "medicine" for a city than to talk with itself, to be open, to change saying:

"I am here for more than 2300 years hosting infinite Stories. Come and create your Story too!"

Who created this project

We are a small team of professionals who live and work in Thessaloniki. We believe that citizens can do more than they think to make their life better. In fact citizens can change their city although the opposite is not true.

This project is a try to give back to our city. It is an official submission for the 2016 [Apps4Thessaloniki competition - Tourism edition](#). We are not affiliated with any public sector organization of the city.

Theodoros Ploumis

Web design and development. Idea development.

Sophia Dampali

Legal advise, terms of use etc.

Frequently Asked Questions

What is this project exactly

Please see the related web page [About](#).

What is a Story

Story is a piece of content up to 400 characters with a title and some fields such as categories, postal address, external link and an associated media content usually taken from the public social media. So imagine the Story as a very small blog post or as a tweet or facebook post with more fields. Stories in this project represent the pebbles in the heart of the city. Many stories create one heart.

Who can create a Story

Everyone can create a Story. You don't have to be a citizen of Thessaloniki or speak Greek. The only requirement is to [create an account](#) on the website and have a Story to say!

How can I browse the Stories

Just go to the [Stories](#) section and you will see the Stories in a list display sorted by date published (latest come first). Furthermore on the bottom of the website there is a bar with many navigation options. When you are on a Story page you can navigate to the Next or Previous Story, open or close the Story details etc. There are some useful keyboard shortcuts to help you navigate on the website (on the website press "?" to get a help popup about keyboard navigation).

How can I search on Stories

You can use the Search button on the bottom bar of the website. Search is powerful, try it and see.

How can I create a user Account

By registering a user account you become a Story Teller that means you can write Stories and manage your own Stories. Creating a user account is pretty simple. The only thing you need is to have an email account and pick a username and a password. You will have to validate your email though to complete your registration. To simplify things we provide also 3rd party account registration through common social network authentication (twitter, Facebook, Google+) so there is no need to add your email and password to login every time.

How can I delete my user Account

You can do this easily through your Account page. Notice that there is a human being behind this process (mainly for security reasons) so your account will not be deleted immediately.

How can I report a Story

If you believe a Story is harmful, spam, offensive etc and/or in generally it does not respect the [Terms of Use](#) you can send as an email to "hi [at] manystoriesoneheart [dot] gr" including the Story

url as also as a description of the issue.

What is "Good" Story to write

First of all the Story must have a (strong) relation to Thessaloniki.

Please, use only English language so more people can read your Story.

Each Story will be shared among people.

Be imaginative, polite and love your Story.

Respect copyright policy of 3rd party content you may use.

Avoid copy-pasting text directly from websites if it is not allowed to. Rewrite texts if so.

There is no restriction to avoid promoting business or other for-profit organizations or events but bear in mind that your Story has to follow the general achievement of this project which can be found in details on the [About](#) page.

Your Story may stay on the history of the city too so enjoy it and take this chance seriously. The best way to write a good Story is to browse the website before writing.

What oEmbed services can I use

We are using [Embed.ly](#) service to provide a wide range of external media providers. Through the [oEmbed](#) protocol we respect content of the media providers by doing a "Fair use". So in fact we support all the [oEmbed providers](#) (more than 300) that are supported by the Embed.ly service.

How can I contribute, report a technical issue etc

Please see the related [Developers](#) page.

How can I reuse the Stories

We put a lot of effort in reusing and sharing Stories of this website. There is a Restful API, RDF, SPARQL Endpoint, RSS and of course the Website that allows the content to be shared. All the Stories are published under a Public Data license which is included on the HTML of each page or on the Restful and SPARQL endpoint accordingly. If you want to create an App and get it listed on this project follow the instructions at github.com/manystories/apps.

How to use the Bookmark

On the bottom of the website there is a [bookmarklet](#) which is a bookmark that does an action on the browser. Drag this bookmarklet on your bookmarks and every time you click on while viewing a webpage you will be redirected to the [add new Story](#) form filled in with content from that webpage. It is like a shortcut that makes it easy to create a new Story from any website.

Which device should I use to have the best experience

We tried a lot to offer an excellent experience for every device and for every browser that someone can use. Being responsive by default the website can be accessed by any known web capable device such as mobile, tablet, laptop etc. Not only the content but also the add new Story form etc. We could say that on tablet devices you may have the greatest experience but this is because the 3rd party (embedded) content such as images - which is usually coming from social media - is created and used for smaller size screens.

Media - Screenshots

Find screenshots at manystories.readthedocs.org/en/latest/media

Developers

This section contains some useful information about the building of this web app.

There is also a [FAQs](#) session you can read before coming here.

Technology used

1. Drupal

[Drupal](#)

The whole web portal is made with Drupal 7.x. There is an HTML5 responsive design built as a subtheme of [Mothership](#).

2. oEmbed

[oEmbed](#)

We use the oEmbed format to "embed" third party media on it. This allows for "fair use" of that media.

3. Embed.ly

[Embed.ly](#)

This is a paid service with free plan available that provides better oEmbed integration.

4. RDF and RSS

We use [RDF](#) to provide linked and structured data for the [Stories HTML](#), for the [SPARQL Endpoint](#) as also as an [RSS feed](#).

We mainly use namespaces such as schema.org and [Dublin Core](http://www.dublincore.org/) for the [RSS](http://www.rss.org/). Here is the complete list of the namespaces that are used:

- [content](#)
- [dc](#)
- [foaf](#)
- [og](#)
- [rdfs](#)
- [sioc](#)
- [sioct](#)
- [skos](#)
- [xsd](#)
- [owl](#)
- [rdf](#)
- [rss](#)
- [site](#)
- [schema](#)

5. HybridAuth

[HybridAuth](#)

We use HybridAuth for easy social sign-in on the portal using accounts from various social apis and identities providers such as Facebook, Twitter and Google.

6. jsonAPI

[jsonAPI](#)

We use this special json specification to create our [RESTful API responses](#). JsonAPI is designed to minimize both the number of requests and the amount of data transmitted between clients and servers. This efficiency is achieved without compromising readability, flexibility, or discoverability.

7. Mousetrap

[Mousetrap](#)

We use this tiny standalone javascript library to create some useful navigation shortcuts.

8. Wikiwand.com

[Wikiwand.com](#)

Wikiwand is probably the best reader for [Wikipedia](http://www.wikipedia.org/). We don't use it as a service but we like it a lot and most of our external links are going to Wikiwand instead of the 'ugly' Wikipedia.

9. Markdown

Markdown

Markdown is a text-to-HTML conversion tool for web writers. We use this tool to create the content for the basic website pages as also as for the Github documentation.

10. jQuery countTo]

jQuery countTo

jQuery countTo is a jQuery plugin that will count up (or down) to a target number at a specified speed, rendered within an HTML DOM element.

11. SPARQL

SPARQL Query Language

There is a whole page about SPARQL.

Accessibility

This project was made with accessibility in mind. Currently it is valid under WCAG 2.0 (Level AA) specification except from the external data that are embeded to the web pages using the oEmbed protocol. For these external date there is no option (technically) to fix any accessibility issues if exist.

Tools used

Sharing here some online tools that helped us create this project.

- HTML Character entity reference Chart
- Bookmarkleter (Create bookmarklet from js)
- Patterncooler.com/ (Free seamless pattern backgrounds)
- ezgif.com (Animated gif editor and gif maker)
- realfavicongenerator.net (A tool to generate favicons, Windows 8 Tiles, Apple Touch icons, Android and iOS icons.)

Icons and other resources

We are using some icons or other media from these providers, creators.

- Slim Square Icons - Basics by icons.design
- Social Media icons by roundicons.com
- licenses.opendefinition.org (json lists of all available open licenses perfect for our open data)
- Accessible - Fat finger form controls (css code)

Issues

You can send us issues or things you believe that need improvement here:

- github.com/manystories/drupal/issues

Contribute

You can create pull requests or explore the project. It is hosted on Github and most of its parts are open to everyone.

- github.com/manystories

RESTful API

Information for how to use the API can be found on the related page: [Restful API](#)

SPARQL Endpoint

Information for how to use the SPARQL Endpoint can be found on the related page: [SPARQL Endpoint](#)

Examples of Apps using the APIs

There are examples and demos available at manystories.github.io/apps/

REST API

About

This is a public available REST API so everyone can consume the data freely. Here are some basic information about the [api](#) for your interest.

- Only Stories are exposed at the moment.
- The data are read only.
- We do not expose any parsed oEmbed results on the API to avoid legal issues with the oEmbed protocol. Instead, we provide only the url of each media file that is used on each Story.
- Each item (Story) is provided under an open source license that is common for all items. Please read the license before use each item.
- Default format of the API is the [jsonAPI](#) but this could change.
- API is under version control (current version is **1.0**). Future versions may be available.
- Default pager shows 50 items. You can override this through a parameter.
- Datetime fields are displayed using the [UNIX time format](#).
- Most of the times results are cached on the server so new Items may not be available immediately after they get created.
- You can create any websites, web or mobile app or anything else using this API with the only obligation to respect the license.
- Do a fair use and respect the systems behind this API.
- By using this API you also agree with the generic [Terms of Use](#).

REST options

By default this app provides a clean read-only http RESTful API under the [api](#) path.

The API is made with the [best practices](#) in mind.

Here are some basic examples of usage. Notice that you may have some programming experience before using the API.

We recommend you use a browser REST client to inspect the API such as [Postman](#), [Advanced Rest Client](#), [RESTClient](#) etc.

1. Discover the resource

Using an HTTP GET request, you can return the Discovery of the resource. The resource will respond with a JSON object that contains useful info for each resource defined by the API. The data results are stored in the data property of the JSON response, while the self and next objects contain information about the resource.

```
curl GET http://manystoriesoneheart.gr/api/
```

2. Get all the available Versions of the resources

By default the API will return the latest versions for each resource. If you want to see all the versions available use the parameter `?all=true` on the request.

```
curl -X GET -i http://manystoriesoneheart.gr/api/?all=true
```

3. Get the Stories resource

```
curl -X GET -i http://manystoriesoneheart.gr/api/stories
```

which is the same as:

```
curl -X GET -i http://manystoriesoneheart.gr/api/v1.0/stories
```

since v1.0 is the latest for the Stories resource.

This will return a list of Items (Stories) with many fields. These fields do match the html display of each Story.

4. Change the Items per page

By default there are 50 Items per page. You can override this on the query.

```
// This will return 2 items per page
curl -X GET -i http://manystoriesoneheart.gr/api/stories?range=2
```

5. Filter the results to get back only specific Fields

```
curl -X GET -i
http://manystoriesoneheart.gr/api/stories?fields=label,universal_id,story_text,media_url
```

6. Apply a Query filter to search within results

Using this option you can filter results using an operator such as "=", "!=", "<", ">" etc.

Here is a list of available operators (if applicable for the specific field type):

```
=
!=
>
<
>=
<=
<>
IN
AND
BETWEEN
CONTAINS
IN
LIKE
NOT IN
STARTS_WITH
```

```
// Get all Stories that belong to the category "Events". Notice that by default the
operator "=" is used.
curl -X GET -i http://manystoriesoneheart.gr/api/stories?filter[categories]=Events

// Get all Stories that belong to the category "Events" AND "Activities". For both
filters we provide the operator "=".
// Do not use simple (') or double quotes (") for the values!
// For example, this is wrong: *filter[categories][value][0]="Events"*

curl -X GET -i
http://manystoriesoneheart.gr/api/stories?filter[categories][value][0]=Events&filter[categories][value][1]=Culture&filter[categories][operator][0]="="&filter[categories][operator][1]="="

// Get the Stories with publish_date=1449764174 OR publish_date=1449953170.
curl -X GET -i
http://manystoriesoneheart.gr/api/stories?filter[publish_date][value][0]=1449764174
```

```

&filter[publish_date][value][1]=1449953170&filter[publish_date][operator][0]="IN"&filter[publish_date][operator][1]="IN"

// Get all Stories with publish_date > 1449764174.
curl -X GET -i
http://manystoriesoneheart.gr/api/stories?filter[publish_date][value][0]=1449764174&filter[publish_date][operator][0]=">"

// Get all Stories that CONTAIN the string "book" on the title.
curl -X GET -i
http://manystoriesoneheart.gr/api/stories?filter[label][value]=book&filter[label][operator]=CONTAINS

// Get all Stories with the media_url field from youtube.
curl -X GET -i
http://manystoriesoneheart.gr/api/stories?filter[media_url][value]=https://www.youtube.com&filter[media_url][operator]=STARTS_WITH

```

7. Get multiple Stories with one request

```

// Load 2 Stories from their universal unique ids
// id1 = 0876712b-186d-443b-8dd0-00ab3c060f05
// id2 = 0587f5e9-175d-4c17-9e90-dcb1204f084f
curl -X GET -i
http://manystoriesoneheart.gr/api/stories/0876712b-186d-443b-8dd0-00ab3c060f05,0587f5e9-175d-4c17-9e90-dcb1204f084f

```

8. Get a Story with an alternative unique id on the path

Notice that this will return the first only result if there are multiple items found.

```

// Get the first Story found with the value "Taste" on field "see_more_text"
curl -X GET -i
http://manystoriesoneheart.gr/api/stories/Taste?loadByFieldName=see_more_text

```

9. Sort by another field (default by publish_date ASC)

```

// Sort by Title, ASC
curl -X GET -i http://manystoriesoneheart.gr/api/stories?sort=title

// Sort by publish_date DESC (latest first).
// Notice the minus sign ('-') in front of the field name in the sort parameter.
curl -X GET -i http://manystoriesoneheart.gr/api/stories?sort=-publish_date

```

REST examples

Here are some very simple examples of how to consume the API. Full examples can be found on

[Github](#).

1. Using jQuery ajax

```
(function($){
  $(document).ready(function () {
    // The main api url for Stories.
    var $url = 'http://manystoriesoneheart.gr/api/stories';

    $.ajax({
      type: 'GET',
      url: $url,
      success: function (results) {
        $.each(results.data, function(index, element) {
          console.log(element);
        });
      }
    });
  });
})(jQuery);
```

2. Using php

```
// REST API url.
$url = 'http://manystoriesoneheart.gr/api/stories';

// Get json response.
$json = file_get_contents($url);

// Decode json to php array.
$stories = json_decode($json, true);

// The array is ready to use.
print_r($stories);
```

SPARQL Endpoint

About SPARQL

Another advanced method to get the Stories from the website except from the [RESTful API](#) is to use the public [SPARQL](#) Endpoint under [/sparql](#).

On the same url there is an interface available which implements [SPARQL](#) and [SPARQL+](#) via [HTTP Bindings](#).

SPARQL is an RDF query language, that is, a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework ([RDF](#)) format. It is one of the key technologies of the semantic web.

SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

We use this technology not only to expose the Stories to the public but also to offer completely Linked data to everyone interested.

Public SPARQL Endpoint

The public SPARQL Endpoint can be accessed at url [/sparql](#).

Here are some examples of SPARQL Queries you can run on the endpoint.

If you are just starting with SPARQL and Linked data here we propose a really good article to read: [Using SPARQL to access Linked Open Data](#)

1. Get the first 100 triples for Pages and Stories

```
SELECT * WHERE {
  GRAPH ?g { ?s ?p ?o . }
}
LIMIT 100
```

2. Get the 10 latest Stories that have the category (keywords) "Gastronomy"

```
PREFIX schema: <https://schema.org/>

SELECT ?Story WHERE {
  GRAPH ?graph { ?Story schema:keywords
<http://manystoriesoneheart.gr/stories/gastronomy> . }
}
LIMIT 10
```

3. Get the latest 10 Stories that have an Event date and happened on September 2015 (01 - 31 Sept. 2015 UTC)

```
PREFIX schema: <https://schema.org/>

SELECT ?Story WHERE {
  GRAPH ?graph {
    ?Story ?has ?startDate .
    ?startDate schema:startDate ?value .
    FILTER(?value >= "1441065600" && ?value <= "1443657600")
  }
}
LIMIT 10
```

4. Get the 10 latest Stories that have the word "Remember" as a call to action

```
PREFIX schema: <https://schema.org/>

SELECT ?Story WHERE {
  GRAPH ?graph {
    ?Story schema:potentialAction ?actionText .
    FILTER(?actionText = "Remember")
  }
}
LIMIT 10
```

Many Stories, Many Apps

A guide to create Apps using the ManyStories project APIs.

This is the repository for the Apps that are using the APIs and data from manystoriesoneheart.gr.

There is a website for these Apps at: manystories.github.io/apps.

You can create an App too and get it listed here.

License of the Data

All the data are available under [Creative Commons Attribution 4.0](https://creativecommons.org/licenses/by/4.0/) but this can change. In any case the license will be an Open license and it will be compatible with the [CKAN](https://ckan.org/) platforms. We are using licenses from licenses.opendefinition.org.

Furthermore the license can be found on the API resource of each item ([example](#)).

License of the Apps

You must have an **open source license** for your App. If you are not sure about the license take some help at choosealicense.com. The license must be included on your App folder as txt or md file.

How to submit an App

You can make a **pull request** on this repository. Of course you need to fork it first.

Are there any html markup that need to contain in a web App

You are free to create your App as you prefer. There are no any styling guidelines etc. Though it would be fair to contain the information below on the top, sidebar or bottom of your App which are:

- The source code of the App (pointing on the app subfolder on repo github.com/manystories/apps)

- A valid title
 - The license of the App as link
 - The logo of the website linked to the main website
-

How to install the Drupal profile

Before starting check the [Drupal documentation guides](#) at Drupal.org.

You may need to have basic programming knowledge.

```
// Download drupal on the folder of your website (let's call this www)
cd www

// The best method is using Drush.
drush dl drupal-7.41

// If you don't have drush
wget https://ftp.drupal.org/files/projects/drupal-7.41.zip
unzip drupal-7.41.zip

// You will now have this structure
// 'www/drupal-7.41/'

// Remove the sites/all folder
cd drupal-7.41
rm -r sites/all

// Clone this repository while being at root/drupal-7.41
git clone git@github.com:manystories/drupal.git

// Move drupal/sites/all folder into sites/all
mv drupal/sites/all sites/

// Install Drupal normally. Select the minimal profile.
...

// On you new website enable the backup_migrate module
drush en -y backup_migrate

// Go to the backup and migrate settings
(admin/config/system/backup_migrate/restore) and restore the demo database found
below.
```

Demo database

- [Download](#) the database.

Drupal admin

- user: admin
- pass: admin

Content

By default only one Story is provided on the demo and all the other pages are empty pointing to the online documentation.

Dashboard

After installing the demo database go to the site Dashboard (</admin/dashboard>) and add/edit settings.