

TUTORIAL - CREATING THE APP 'MY CITY, MY PLACE' ON MICROPASTS

Deep Cities Project

Authors: Elisa Broccoli, Chiara Bonacchi e Marta Krzyzanska

INTRODUCTION

Pybossa technology for crowdsourcing

In this tutorial, we will consider how to enable the crowdsourcing of research data using the Pybossa framework (<http://pybossa.com>), a technology to collect, enrich and analyse data via small tasks submitted by relatively large groups of people, over the Internet. Pybossa was developed by SciFabric under the technical lead of Daniel Lombraña González (<http://github.com/pybossa/pybossa>).

To power the crowdsourcing of research data through Pybossa, there are three possible routes that can be followed. The first is to install your own Pybossa server and develop a bespoke crowdsourcing website, following the documentation available here: <https://docs.pybossa.com/installation/gettingstarted/>. This is the route chosen by the MicroPasts team and also by the British Library, whose LibCrowds initiative is powered by Pybossa (<https://www.libcrowds.com>). The second route is to create an account in the Pybossa server, and then a project that runs on <https://scifabric.com/crowdcrafting/>, by following these steps: <https://docs.pybossa.com/build/overview/>. The third route is that of collaborating with a themed crowdsourcing website built using the Pybossa framework, such as MicroPasts (<http://crowdsourced.micropasts.org>). This tutorial will focus on this third and last route.

The MicroPasts crowdsourcing platform

The MicroPasts crowdsourcing platform was built as part of the MicroPasts project, funded by the UK Arts and Humanities Research Council and developed as a collaboration between the UCL Institute of Archaeology and the British Museum. The platform is now maintained by the main MicroPasts team (Andy Bevan, Daniel Pett, Chiara Bonacchi, Jennifer Wexler), together with other institutions, researchers and heritage professionals internationally.

The MicroPasts crowdsourcing platform runs on the Pybossa framework developed by SciFabric. Templates have been created so far by the project team and by collaborators, by elaborating on and modifying some of the templates already available in Pybossa. The MicroPasts templates are modular, this means that they are adaptable to the needs of different collections and institutions. So far the kinds of tasks that are supported by the MicroPasts templates are: transcription of structured and unstructured text and of sound; geo-referencing of find spots; classification of photographic material; video-tagging; and 3D photo-masking.

Your City, Your Place App in MicroPasts

In this tutorial you will learn the technology and workflow behind cloning and adapting an existing crowdsourcing template in MicroPasts, through the example of the 'Your City, Your Place' app template, which was created in 2021 for the Deep Cities Project, which studies sustainable urban transformations through heritage. The participants' memories will be part of a digital inventory of the case study's area that will be shared with urban planners and heritage experts. Through this app, the participants' voices and values will contribute to future decisions in their city's development and the maintenance of its heritage.

The 'Your City, Your Place' App collects current experiences and past memories related to a selected urban area. It loads a map of the area, and invite participants online to find a place or building that is significant to them and add a marker to indicate its position- Then, the app invites participants to write in the Memory Box why the place is significant to them, for example by sharing a story.

PART 1 - CLONING AND ADAPTING THE "YOUR CITY, YOUR PLACE" APPLICATION ON GITHUB



1. Setting-up GitHub

The crowdsourcing application you will develop will be accessible via the MicroPasts crowdsourcing website (<http://crowdsourced.micropasts.org>). The code will be created and made publicly available via the MicroPasts GitHub account (<https://github.com/MicroPasts>). GitHub is a platform for collaborative coding which enables version control. This means that all the changes that are made to the code as you go along in your coding practices are tracked; it also means you can ask collaborators for help to solve issues and clone and modify copies of the repositories that already exist on a project account and are publicly shared.

The first thing to do, in order to clone and modify an application via MicroPasts, is then to set-up a GitHub account. To do this, go to <https://github.com>, sign-up for an account and log into it. To code using Git, you can either run commands directly via your terminal, or use a

desktop version of GitHub. In this tutorial we will choose the second route, so you will need to download and install the version suitable for you, macOS or Windows, of GitHub Desktop available at <https://desktop.github.com/>.

2. Clone an existing "Your City, Your Place" application

Once you have installed GitHub Desktop, you are ready to clone the code of an existing application, so that you can then modify it and adapt it for your research.

To clone the code of an existing application, go to the MicroPasts project page on GitHub <https://github.com/MicroPasts>. Here, you will find all the code that powers MicroPasts, shared in different and dedicated repositories. You will need to find the repository (repo) that contains the code of a recent application similar to the one you would like to develop.

For this tutorial you will find the code of the 'Your City, Your Place' application on Dr Elisa Broccoli's GitHub. Go to her page on GitHub <https://github.com/E-Broccoli> and search for the repo 'Deep-Cities_Your-City-Your-place', which contains the code for an application to mapping the stories and memories of the New Waverley urban area in Edinburgh.

Once you have found the 'Deep-Cities_Your-City-Your-place' repo, click on it so you visualize the webpage https://github.com/E-Broccoli/Deep-Cities_Your-City-Your-place. Then you will need to click on the green button 'Code' and choose the option 'Open with GitHub Desktop'. When prompted, click on 'Clone' and the repo will be saved locally on your computer in a folder called 'GitHub' (unless you have previously changed the default name).

3. Create a repo for your new application

Once you have cloned a copy of the application 'Deep-Cities_Your-City-Your-place', you will need to create a repo for the new application you are developing. The subsequent step, in fact, will be to copy paste locally, on your computer, all the files contained in the 'Deep-Cities_Your-City-Your-place' repo into the new repo you have created and make the necessary changes. To create a new repo for your application, you need to go to your GitHub page, click on the '+' button on the top right side of the page and choose the option 'New Repository'.

When prompted, choose a name for your repository such as 'Your city, Your Place - (name of your case study's area)'. For this tutorial we will use the name 'Rome' as an example, so the name of the new repo will be 'Your City, Your Place - Rome'. Leave the default viewing option set to 'Public', tick the option 'Initialise this repository with: Add a README file', and then click on 'Create repository'. Ticking the 'Initialise this repository with: Add a README file' option will allow you to create a repo that already has a so-called 'README' file inside, where you will write the metadata that explains what your repo is, who developed it and how, under what licenses and for what uses. Once you have done this, click on 'Code' and choose the 'Open with GitHub Desktop' option. When prompted, click on 'Clone' and the repo will be saved locally on your computer in a folder called 'GitHub' (unless you changed the default name).

4. Copy and modify files in your new repo

Now you should have two repos visible on your GitHub Desktop: 'Deep-Cities_Your-City-Your-place', containing the files that power the application created for the Deep Cities Project's case studies, and 'Your City, Your Place -Rome' which is your new repo that at present contains only a README file with the title of your repo. What you want to do is to go to the folder where the repos are stored locally. So, in your Finder, search for a folder called 'GitHub'. The 'GitHub' folder should contain two folders called, respectively 'Deep-Cities_Your-City-Your-place' and 'Your City, Your Place - Rome' (your two repos). Now copy all the files contained in 'Deep-Cities_Your-City-Your-place' except for the README and paste them inside the 'Your City, Your Place - Rome' folder. Now if you go back to your GitHub Desktop and click on your 'Your City, Your Place - Rome' repo, you should see that GitHub has tracked the additions and changes you have made. This is what version control means in practice: tracking changes made by developers, so they can be reviewed.

Now that you have the files that power a 'Your City, Your Place' application, you will need to adapt them so that they suit the mapping of the memories and stories of your case study's urban area. To know how to do this, you first need to know what files are inside the repo and what they are useful for. The 'Your City, Your Place - Rome' repo contains the following files: 'requirement.txt' (requirements to run the application), 'template.html' (this is useful to enable the view for every task and to deal with the data of the answers), 'tutorial.html' (this is the file to show the tutorial that guides users in the completion of the app tasks), 'README.md' (a markdown file that contains metadata about the repo), 'long_description.md' (the description of the application that will appear on the MicroPasts crowdsourcing website), 'project.json' (it contains the setup metadata) and 'AUTHORS.md' (a markdown file that contains metadata about the authors of the app).

You will need to modify the following files:

(1) README.md: Open the README file inside the 'Deep-Cities_Your-City-Your-place', copy all the text, and paste it in the README file of the 'Your City, Your Place - Rome' folder under the text that is already there. Save and close.

(2) AUTHORS.md: Change the name and email address to your name and email address. Save and close.

(3) Long_description.md: update the text to suit your research and your case study area. You need to describe your application, its aims, and what are the tasks of the participants. You can also include the URL of the logos of the funded partners involved, so that they are displayed. Save and close.

(4) Project.json: Change "name" and tape "Your City, Your Place - Rome"; change "short_name" and tape "(name of the case study's area)", so in this tutorial "rome"; and change "description" and "question". Save and close.

(5) Template.html: This file allows users to view the interface of the 'Your City, Your Place' application, so they can submit their answers. To edit this file you need to use an external editor like Atom <https://atom.io/>. Once you have downloaded the editor, you need to right click the

template file in GitHubDesktop and click 'Open in external editor'. It is necessary change some details in the code to adapt the app to your case study.

1- change the location coordinates: click CTRL+F and digit "setView", then you need to substitute the existing coordinates with the Latitude and Longitude (in this order) of your case study's area. The app uses OpenStreetMap <https://www.openstreetmap.org>. To find the coordinates you need to digit the location of your area in the search bar of the OpenStreetMap's website, and drag the map to get the exact view you want the app will show. Once you are happy with it, copy the coordinates prompted in the URL bar. For instance, in the URL "<https://www.openstreetmap.org/#map=18/55.95181/-3.18121>", the Latitude is 55.95181, and the Longitude "-3.18121".

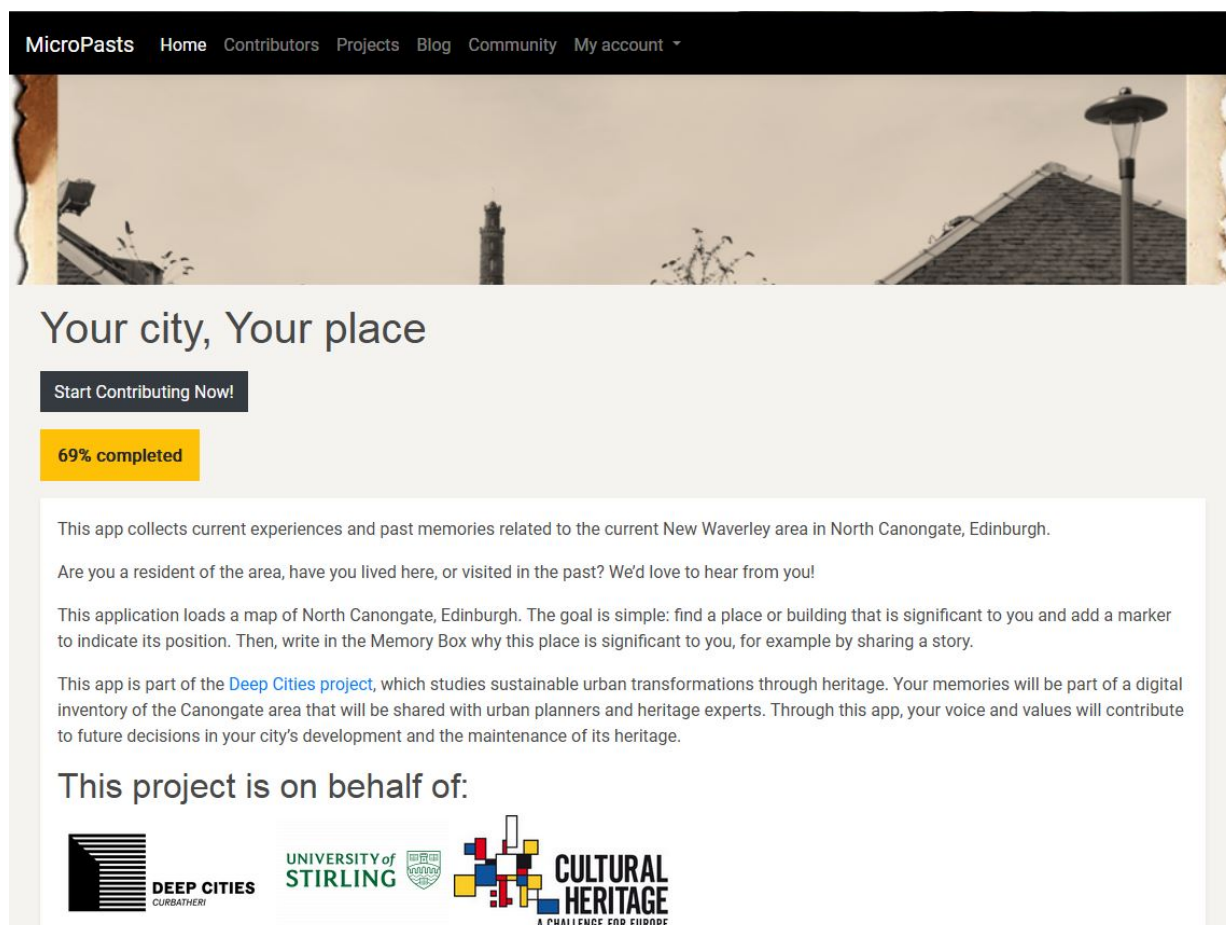
```
var Latitude;  
var Longitude;  
var Marker;  
var Connections;  
var map = L.map('map',{dragging:false, scrollWheelZoom:false, zoomControl: false}).setView([55.951957, -3.181014], 17);  
L.tileLayer('https://{s}.tile.osm.org/{z}/{x}/{y}.png', {  
  attribution: '&copy; <a href="https://osm.org/copyright">OpenStreetMap</a> contributors'  
}).addTo(map);
```

2- project name: Substitute 'Urbplan' with 'rome' every time it features (CTRL+F help you to find a word quickly through the code). Save and close.

(6) Tutorial.html: Adapt the existing tutorial to your app or delete and write a new one using the external editor. Save and close.

Now go to your GitHub Desktop and click on your 'Your city, Your place - Rome' repo. All the changes you made should be showed. Now you will need to update the online version of the repo with all the additions and changes you made to your local version of the repo; in Git this is called 'committing' changes. To commit your changes, just write a Summary and a Description describing the changes and then click on the 'Commit to main' button at the bottom of the page. Then press 'Sync' (in some versions of GitHub this is 'Push commits to the origin remote' 'Push Origin') at the top of the page. The online version of your repo should now be exactly the same as the local copy of 'Your City, Your Place - Rome' that you have locally on your computer, in the 'GitHub' folder.

PART 2 - SETTING UP A NEW PROJECT ON MICROPASTS



MicroPasts Home Contributors Projects Blog Community My account ▾

Your city, Your place

Start Contributing Now!

69% completed




This app collects current experiences and past memories related to the current New Waverley area in North Canongate, Edinburgh.

Are you a resident of the area, have you lived here, or visited in the past? We'd love to hear from you!

This application loads a map of North Canongate, Edinburgh. The goal is simple: find a place or building that is significant to you and add a marker to indicate its position. Then, write in the Memory Box why this place is significant to you, for example by sharing a story.

This app is part of the [Deep Cities project](#), which studies sustainable urban transformations through heritage. Your memories will be part of a digital inventory of the Canongate area that will be shared with urban planners and heritage experts. Through this app, your voice and values will contribute to future decisions in your city's development and the maintenance of its heritage.

This project is on behalf of:

1. Setting-up a MicroPasts Account

To create a crowdsourcing project you need to be a registered user on the <https://crowdsourced.micropasts.org> site (to register go here: <https://crowdsourced.micropasts.org/account/register>). You must then sign in to the site and find out your personal API key which is available from user dropdown menu on the top right of the page ('My Profile').

2. Create a new project

You can create a new project from scratch or more likely modify an existing template of a project and add this to the site as a new project. Most of the time it is easiest to copy the relevant files from an existing project (on MicroPasts or Crowdcrafting for instance), modify them a bit to get the style and content you want, then use them to create a new project.

To create a blank project, making sure you are logged in, and then go to <https://crowdsourced.micropasts.org/project/new> Then in the resulting dialog, we add the 'Name' of the project such as "Your city, Your Place - Rome", a 'short name' for the project - that must have no spaces, as it will go in the project URL, for example "rome" or if this is already taken, some other short name that has no spaces in it - and a longer description (which is the project abstract or description and which can be written in plain text or with markdown elements to add institutional logos, etc.). For this example use the items created before for the Project.json file in your GitHub repo, just copy and paste. Once you have added these, click Create.

This creates a new project on the MicroPasts site and the resulting dialog allows you to further modify it, for example adding a project avatar (i.e. a picture/logo), assign it to an existing category of project (only admin users can create new categories however), and then Save the Changes. You should then be taken to a new page where you can see your avatar, project name and description (you can always go back and edit the above details at: <https://crowdsourced.micropasts.org/project/rome/update>. Please substitute the name 'rome' with the short name of your project in the URL).

3. Create a Task Presenter

You now need to create the Task Presenter which is the page that loads and shows a new task to the end user. This can be found from the left-hand menu Tasks-Task Presenter (or directly <https://crowdsourced.micropasts.org/project/rome/tasks/taskpresentereditor>. Please substitute the name 'rome' with the short name of your project in the URL). In fact, you always start with a template, even if just a very basic one. For this example, click on 'Basic' and in the resulting dialog, you will see the html and javascript that would create the page the users sees on starting a task. In our current example, we will simply delete everything in the current html box and paste in everything in template.html file in your GitHub repo. Then click 'Update'.

You will probably need to go back to fine-tune the html of the Task Presenter later and can always do so by going back to the Task Presenter page at <https://crowdsourced.micropasts.org/project/rome/tasks/taskpresentereditor> (Please substitute the name 'rome' with the short name of your project in the URL). However, for now the above should allow you to create some dummy tasks and check the overall functioning of the project.

4. Import Tasks

Now you can go to your project: <https://crowdsourced.micropasts.org/project/rome> (please substitute the name 'rome' with the short name of your project in the URL), then to 'Import Tasks - Import'. Click on 'Import Data below 'Local Drive CSV Spreadsheet'. Now upload the file 'cities.csv' that is in the 'Deep-Cities_Your-City-Your-place' folder in your computer. Click 'Import'.

5. Test and Publish your project

Now it should load the tasks. And when you click on 'TestIt!' top right it should load the way the user would see the task. You then tinker with the Task Presenter until you are happy with it and the other set-up. Then delete the example task and load the real ones, and then 'Publish' the project. If you wish you can also ask an admin user to move it into the 'Featured Projects' section of the MicroPasts site to increase its prominence.