D- Light Content Management Guide

Welcome to the content management guide for your website. This document is designed to help non-technical users update and manage the content within the "content" directory of the website. Please follow these instructions carefully to ensure that your website remains functional and up to date.

Important Practices & Tips

Backup Before Changes: Always make backups before making changes.

Consistent Naming: Follow a consistent naming convention for images, PDFs, audio files, JavaScript (.js) & JSON (.json) files to keep your project organized.

Validation and Testing: After updating your files, test your website to ensure the new or edited material is displayed correctly.

Document Changes: Keep a record of changes made to personas, especially if working in a team.

By following these guidelines, you can manage personas, languages & activities on your website effectively, ensuring that the content is always up to date and accurately reflects what you wish to showcase.

Managing Audio Content

For Personas' Life Stories (‘audio’ Directory)

The audio directory contains audio files that narrate the life stories of personas represented on the website. Here’s how to manage these files:

Navigate to the audio Directory: Inside the content folder, you will find an audio folder. This folder contains subfolders for each language (en for English, sv for Swedish).

Audio File Format: Ensure your audio files are in MP3 format. This is the required format for compatibility with web browsers.

File Naming Convention: The name of the audio file must follow the format Name-audio.mp3, where Name is the name of the persona with the first letter capitalized. For example, if you have a persona named "Emily," the file should be named Emily-audio.mp3.

Adding a File: To add a new audio file, place it in the correct language subfolder. For example, if adding an English audio file for Emily, it should go in content/audio/en/Emily-audio.mp3.

Adding New Languages: If you introduce a new language, create a new folder within the audio directory with the appropriate two-letter language code (e.g., et for Estonian) and place the corresponding audio files inside it.

Note that adding a new language requires updates in other parts of the site as well (discussed later in this document).

For Daily Audios (‘audioToday’ Directory)

For audio files related to current events or daily updates, you will use the audioToday folder. The process is the same as the audio directory:

File Naming: Use the same naming convention, Name-audio.mp3.

Placement: Put the audio files in the respective language subfolder.

.

Managing Image Content

For Personas (‘img’ Directory)

Location: Find the content/img directory in your project folder.

Image Format: The images should be in JPG format for best compatibility.

File Naming: The name of the image file should be the lowercase name of the persona followed by .jpg. For example, for a persona named "Emily," the image file should be named emily.jpg.

Adding an Image: To add an image, simply place it in the img directory.

Image Size: For best performance and layout, ensure the image sizes are consistent. You may need to resize or crop images to fit the design of the site.

General Notes

File Names: The naming convention is crucial. The system is case-sensitive and expects an exact match for file names.

Audio and Image Synchronization: Ensure that each persona's audio files and image files have consistent naming. This will link the audio and images correctly on the website.

Adding New Content: When adding new content (audio or images), ensure that you're not replacing existing files unless you intend to update them.

Backup: Before making changes, create a backup of the existing content directory in case you need to revert to the original state.

Testing: After making changes, always test the website to make sure the new content is displaying and functioning correctly.

New Languages: Introducing a new language involves more than just adding a new folder in audio; you'll need to update language settings across the website, which may require technical assistance.

This guide should empower you to manage the content of your website effectively. For substantial changes or if you encounter issues, consider seeking assistance from a web developer.

Managing Language Files & Changing the Text

The lang folder contains JSON files that store the text for different languages on your website. These files allow you to maintain multilingual content easily. Here's how to manage and modify the language files:

Understanding the JSON File Structure

JSON files are structured in a key-value pair format, making it easy to locate and update text. Here’s a breakdown of the structure:

Key: It's a unique identifier that the website code uses to fetch and display the correct text. For example, "headerHome": "Home" means headerHome is the key for the homepage title text.

Value: It's the actual text that will appear on your website. In the example above, "Home" is the value that will be displayed for the headerHome key.

Editing an Existing Language File

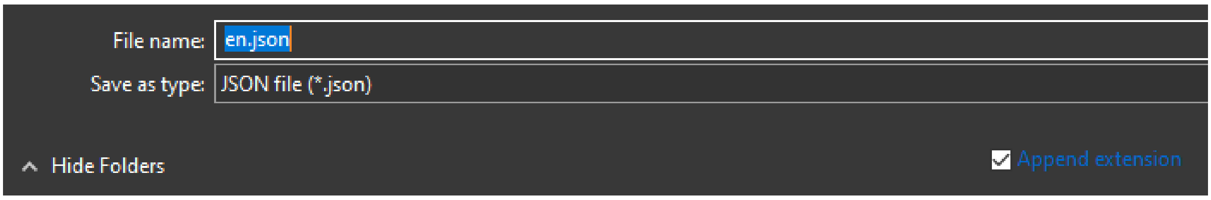
To change text on your website for a particular language:

1. **Open the Language File:** Navigate to the lang folder and open the respective JSON file, such as en.json for English.

2. **Find the Key:** Locate the key corresponding to the text you wish to change. Use the key naming convention to help you find the right one. For instance, lifeSection1Banner is the text on the life page, on section one on the banner.

3. **Edit the Value:** Modify the value next to the key to update the text. Ensure to maintain the quotation marks and the comma at the end of the line, if present.

4. **Save the File:** After making changes, save the file. Ensure the file maintains the .json extension and the correct language code as its name. When saving, the .json file extension might not be present, in this case choose “All types” or “Any type” and manually write .json in the name.



File name: en.json

Save as type: JSON file (*.json)

☑ Append extension

Editing Text with HTML Tags

A small amount of values may have HTML tags inside the JSON values, these are needed if the text for example has clickable hyperlinks in it. In this case you must be careful not to remove the tags (e.g. <p> & </p>) and only change the text inside them.

For example:
"footerHTMLCredit": "<p>Developed by <a href=https://prakticum.fi/'>Prakticum</a></p>",

In this line, <p> is a paragraph tag, and <a href='https://prakticum.fi/'> is a link tag with an attribute for the URL.

When editing text that includes HTML tags:

Identify the Editable Text: Only change the text content and not the HTML tags themselves. In the example above, you would change "Developed by" but leave the <p>, <a href='...'>, and </a></p> tags untouched.

Maintain HTML Integrity: Ensure that opening tags have corresponding closing tags. For instance, every <p> should be paired with a </p>.

Edit with Caution: If you are not familiar with HTML, proceed with caution. Editing the attributes of an HTML tag, like the href attribute of an <a> tag, could break the functionality or layout of your page.

Avoid Injecting Dynamic Data: Do not insert untrusted content into your HTML tags, as this can pose a security risk (such as an XSS attack).

Adding a New Language

To add a new language to your website:

Duplicate an Existing File: Copy an existing JSON file, such as en.json, and rename it to the new language code (e.g., fi.json for Finnish).

Translate the Content: Open the new JSON file and translate the values into the new language. Do not change the keys.

Add Language Code: You'll need to add the new language option to your language switcher in the HTML. This typically involves adding a new button or selector for the new language. Contact your IT department with software knowledge to successfully implement the feature.

Tips for Managing Language Files

Use a Text Editor: Edit the JSON files using a plain text editor that supports JSON syntax highlighting, such as Notepad++ or Visual Studio Code.

Formatting: JSON files must be correctly formatted to work. Each key and value should be enclosed in double quotes, and each key-value pair should end with a comma, except for the last pair.

Comments: JSON does not support comments. If you need to leave notes, you might need to create a separate documentation file.

Validation: After editing a JSON file, using a JSON validator tool to ensure there are no syntax errors might be needed.

Backup: Always keep a backup of the original files before making changes.

Testing: After updating a language file, test your website to make sure the changes appear correctly and there are no errors.

By following these guidelines, you can efficiently manage the language files in the content/lang folder to update the multilingual content on your website. Remember to document any changes made, including translations and the addition of new languages, to maintain a clear record.

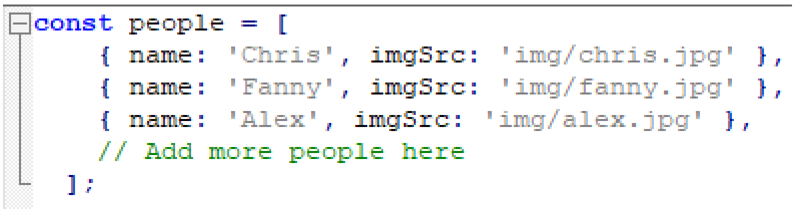
Managing People Content

The people.js file which is found in the “content” folder is crucial for adding and managing the personas showcased on your website. It connects the personas' names with their respective image and potentially other attributes such as audio files. Here's how to effectively manage and update this file.

Adding a New Persona

To add a new persona to your website, follow these steps:

1. **Prepare the Content:**
    - Ensure you have an image for the new persona, named according to the guidelines (all lowercase, .jpg format).
    - If applicable, prepare the audio file for the personas life story and mood, following the naming convention (Name-audio.mp3 with the first letter of the name capitalized).

2. **Update people.js:**
    - Open the people.js file in a text editor that supports JavaScript syntax highlighting. (Notepad works, but an editor which supports syntax highlighting is highly recommended, such as Notepad++)
    - Add a new object to the people array for your new persona. For example, to add a persona named "Alex", copy the line of an existing persona and change the name on the “name” + “imgSrc” to your new persona :
    - 

```
const people = [
    { name: 'Chris', imgSrc: 'img/chris.jpg' },
    { name: 'Fanny', imgSrc: 'img/fanny.jpg' },
    { name: 'Alex', imgSrc: 'img/alex.jpg' },
    // Add more people here
  ];
```

    - Remember to save the file with the .js file extension.

3. **Place the Content:**
   - Upload the image to the content/img directory.

   - Upload the audio file to the appropriate language directory within content/audio or content/audioToday, depending on its purpose.

Removing a Persona

To remove a persona:

Locate the Persona: Find the object in the people array in people.js that corresponds to the persona you wish to remove.

Remove the Object: Delete the entire object for that persona. In the people.js file remove the line where your persona is located, be careful not to remove anything else.

Remove Associated Content: If there are image and audio files associated with this persona, consider removing them from their respective directories unless they are shared with other personas.

Best Practices

Backup Before Changes: Always make a backup of people.js before making changes.

Consistent Naming: Follow a consistent naming convention for images and audio files to keep your project organized.

Validation and Testing: After updating people.js, test your website to ensure the new or edited personas display correctly.

Document Changes: Keep a record of changes made to personas, especially if working in a team.

By following these guidelines, you can manage personas on your website effectively, ensuring that the content is always up to date and accurately reflects the personas you wish to showcase.

Managing Activities in Your Project

The activities_{language}.js files are where you define the activities available on your site, categorized by types such as movement, and tailored for different languages (e.g., activities_en.js for English). Here's a detailed guide on how to manage these files.

Adding a New Activity

To add a new activity to your site:

1. **Open the Correct File:** Navigate to the content directory, and open the activities_{language}.js file for the language you wish to add an activity to. For example, activities_en.js for English.

2. **Find the Right Category:** Activities are grouped by categories like movement, food & entertainment. Determine which category your new activity fits into.

3. **Copy an Existing Activity:** If you're unsure about the format, copy an existing activity within the category as a template. Always paste in your new activity at the top inside the specific category, to avoid syntax errors.

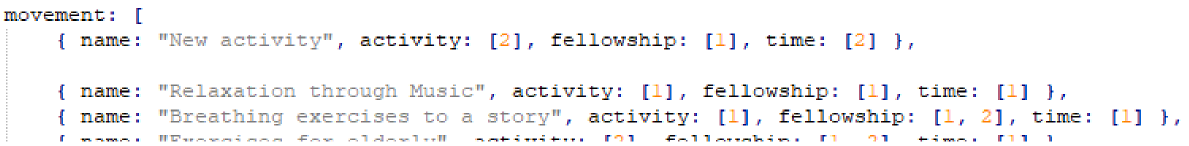
4. **Edit the Activity Details:**
   - **Name:** Replace the name value with the name of your new activity. Keep the naming concise and descriptive.

   - **Activity:** Use **[1]** for calming activities, **[2]** for activating activities, or both **[1, 2]** if it applies to either.

   - **Fellowship:** Use **[1]** for activities done alone, **[2]** for those done together, or both **[1, 2]** if applicable to both settings.

   - **Time:** Use **[1]** for short duration activities, **[2]** for long duration, or both **[1, 2]** if the activity can vary in length.

   Example of adding a new activity:



```
movement: [
    { name: "New activity", activity: [2], fellowship: [1], time: [2] },

    { name: "Relaxation through Music", activity: [1], fellowship: [1], time: [1] },
    { name: "Breathing exercises to a story", activity: [1], fellowship: [1, 2], time: [1] },
    { name: "Exercises for elderly", activity: [2], fellowship: [1, 2], time: [1] }
```

1. **Add Your Activity:** Insert your new activity object into the category array, following the format of existing entries.

2. **Save the File:** After adding your new activity, save the file with the **.js** file extension.

Adding PDF’s to the activities

To add a PDF to a recently added activity:

1. **Location:** Find the content/pdf/”Your-Language” directory in your project folder.

2. **File naming:** Save the PDF with the exact same name as added in the activities_{language}.js file. For example in the above image where we added a new activity called New activity, we would name the PDF exactly (case sensitive) New activity.pdf.

3. **Adding an Image:** To add a PDF, simply place it in the correct language PDF directory.

Removing an Activity

To remove an activity:

1. **Locate the Activity:** In the activities_{language}.js file, find the activity you wish to remove within its category array.

2. **Remove the Activity:** Delete the entire object that represents the activity, including the surrounding curly braces {} and the trailing comma if present.

1. **Save the File:** Save your changes to the file with the **.js** file extension.

Editing an Activity

To edit an existing activity:

1. **Find the Activity:** In the relevant activities_{language}.js file, locate the activity you want to edit.

2. **Make Your Edits:** Change the activity's name, category, or attributes (activity, fellowship, time) as needed.

3. **Save the File:** Save your changes with the **.js** file extension.

Best Practices

Consistency: Maintain consistent naming and formatting within the file to ensure the site functions correctly.

Testing: After making changes, test your site to ensure the new or edited activities appear as expected.

Documentation: Keep a record of changes made, especially if managing multiple language versions.

Backup: Create backups of your .js files before making significant changes.

Adding a New Language

If adding activities for a new language:

1. **Duplicate an Existing File:** Copy an existing activities file, rename it to the new language code (e.g., activities_fi.js for Finnish), and place it in the same directory.

2. **Translate Content:** Translate the activity names and descriptions within the new file. *DO NOT* change the name of the categories.

3. **Integrate the New Language:** Ensure your site's language switcher and loading logic support the new language file. This can easily be done by the help of your IT department or external developers.

By following this guide, you can effectively manage the activities on your site, ensuring a diverse and engaging experience for users across different languages.