|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| zAssessment # and title | | | | C-WT-AT2-POR-Part-1 | | | | Web Technologies Portfolio Phase1 | | | | | |
| **Lecturer name** | | | | *Adrian Gould* | | | | | | | | | |
| **wStudent name** | | | | *Alexander Glover* | | | | | | | | | |
| **Student ID number** | | | | *200085553* | | | | | | | | | |
| **Telephone contact number** | | | | *0438725598* | | | | | | | | | |
| **Email** | | | | *20085553@tafe.wa.edu.au* | | | | | | | | | |
| **By completing and submitting this signed form to my lecturer, I am stating that:**   1. The attached submission is completely my own work 2. I have correctly cited all sources of information used in this work (if required) 3. I have kept a copy of this assessment (where practicable) 4. I understand a copy of my assessment will be kept by the NMTAFE for their records 5. I understand my assessment may be selected for use in the NMTAFE’s validation and audit process to ensure student assessment meets requirements | | | | | | | | | | | | | |
| **Student Signature** | | *AG* | | | | | | | **Date** | | *28/08/2023* | | |
| Assessors please note: Where verbal clarification has been sought from a student to gather additional assessment evidence from an assessment item, question/s and response/s must be recorded, signed, and dated by the assessor, against the relevant assessment item/s. | | | | | | | | | | | | | |
| NB: Feedback will be given via Blackboard when possible. | | | | | | | | | | | | | |
| **Submission 1** | | Result | Satisfactory / Not Yet Satisfactory | | | | | | | Date | | |  |
| *To satisfy requirements for this assessment, you need to complete the following:* | | | Feedback to student… | | | | | | | | | | |
| **Submission 2** | | Result | Satisfactory / Not Yet Satisfactory | | | | | | | Date | | |  |
| *To satisfy requirements for this assessment, you need to complete the following:* | | | Feedback to student… | | | | | | | | | | |
| **Student Feedback** | | | Feedback from student… | | | | | | | | | | |
| Lecturer Signature | | |  | | | | Student Signature | | | | | |  |
| **Assessment type (🗹):** | | | | | | | | | | | | | |
|  | Questioning (Oral/Written) | | | |  | 3rd Party Report | | | | | |  | Practical Demonstration |
|  | Other – Project/Portfolio (*please specify on the right):* | | | |  | | | | | | | | |

Table of Contents

[Assessment # and title 1](#_Toc142411006)

[Lecturer Signature 1](#_Toc142411007)

[Student Signature 1](#_Toc142411008)

[Assessment Steps 12](#_Toc142411009)

[Complete the Front Page 12](#_Toc142411010)

[Setting Up for Development 13](#_Toc142411011)

[Create Project in IDE 13](#_Toc142411012)

[Update Git Ignore 13](#_Toc142411013)

[Commit and Push Code 14](#_Toc142411014)

[What is the name and the URL of your private repository? 14](#_Toc142411015)

[Create HTML for Page 14](#_Toc142411016)

[Validating the HTML 15](#_Toc142411017)

[Add a screenshot of the validation results for your code 15](#_Toc142411018)

[Adding the Provided Content 15](#_Toc142411019)

[Add a screenshot of the validation results for your updated code 15](#_Toc142411020)

[Testing the Web Page 16](#_Toc142411021)

[Which browser is this screenshot from? 16](#_Toc142411022)

[Screenshot 16](#_Toc142411023)

[Which browser is this screenshot from? 16](#_Toc142411024)

[Screenshot 16](#_Toc142411025)

[Which browser is this screenshot from? 17](#_Toc142411026)

[Screenshot 17](#_Toc142411027)

[Commit and Push Code 17](#_Toc142411028)

[Demonstrate to Assessor 17](#_Toc142411029)

[Submission of Portfolio Work 18](#_Toc142411030)

[*All answer documents MUST be submitted in Microsoft Office 365’s Word format.* 18](#_Toc142411031)

[Appendix A: Code Style Guidelines 19](#_Toc142411032)

[PHP Code (General) 19](#_Toc142411033)

[Applications Built with Laravel 19](#_Toc142411034)

[HTML Code 19](#_Toc142411035)

[JavaScript 19](#_Toc142411036)

[Python Code 19](#_Toc142411037)

[JSON Code 19](#_Toc142411038)

[Appendix B: Naming Conventions 20](#_Toc142411039)

[HTML, CSS and JS Files and folders 20](#_Toc142411040)

[File extensions 20](#_Toc142411041)

[Appendix C: Site Folder Structure 21](#_Toc142411042)

[Appendix D: Conventional Commit Messages 22](#_Toc142411043)

[How to Create SCMs/CCMs 22](#_Toc142411044)

[Example 22](#_Toc142411045)

[Types of Commit 22](#_Toc142411046)

[Types of Scope 23](#_Toc142411047)

|  |
| --- |
| Assessment Due Date |
| This assessment is split into components that have several due dates.  This component of the Portfolio of Work is due:   * Week 8 at 17:30 (5.30PM) on day of scheduled lecture   These are also shown in the Learning and Assessment Plan.  Also, refer to Blackboard for most accurate dates, which may alter due to unforeseen circumstances.  We also will endeavour to update these document(s) at the same time. |
| Required Resources |
| The base requirements this assessment task, are listed below. They are listed as common (for both PC and Mac), and for the individual operating systems.  We presume that all assessment work is completed on a **PC** with the software as specified. This is to reduce configuration issues affecting the successful completion of the assessment item.  Whilst other applications and operating systems may be used, we are unable to give extensive support to ensure your environment is working as expected. Common:  * Access to Office 365 & Microsoft Word * WebStorm, PhpStorm * Access to Figma.com for diagrams * Access to TAFE Web Site Hosting (located on dev.ScreenCraft.net.au) * WinSCP for deployment of site * Git for version control, GitHub account for  PC:  * Web Browsers (Must have **at least** TWO different rendering engines)   + Chromium based: Edge, Chrome,   + Firefox   + Opera  Mac:  * Web Browsers (Must have **at least** TWO different rendering engines)   + Chromium based: Edge, Chrome,   + Firefox   + Opera   + Safari   ***Use of some of these items may not be used in this part of the assessment task.*** |
| Optional Requirements |
| An application to provide web services such as web server, database and more.   |  |  | | --- | --- | | Windows | MacOS | | Laragon  AMPPS  XAMPP  Docker | MAMP  AMPPS  XAMPP  Docker |   The lecturers will not be familiar with all the options listed, and it will be up to the student to investigate how to set up and use any alternatives.  The use of alternative IDEs is allowed (such as VS Code), but not supported. |
| Instructions |
| Follow the steps listed in this assessment item.  Submission of the documentation, code, and associated items is at the end of each part of the portfolio.  Provide evidence in the form of scanned documentation, completed assessment documents, screenshots, screencasts and other formats as required in this assessment. |
| Scenario |
| You are currently working as an intern for a small Perth-based start-up company called *Incredibly Obvious Technologies*.  The company is challenging you to work on a project to assess your skills as you develop them.  The project will build up in phases from the base content by adding semantic HTML, CSS to provide layout and images to provided content.  The final section of the project is to add JavaScript to solve a specific problem.  You will find the project specifications, an outline of the phases of development and other details are given below. |
| Portfolio Specifications and Development Outline |
| The following are the specifications for the project you are undertaking as part of this portfolio of work.  Each part of the project corresponds to another part of the portfolio, with each part submitted according to provided deadlines.   * Phase 0:   + Setting Up – not assessed but contains the required steps to initialise the portfolio. * Phase 1:   + Implement and test semantically and structurally correct HTML for the provided layout design and provided content. * Phase 2:   + Implement and test the required CSS to provide the layout as provided. * Phase 3:   + Add additional content and styling to amend the project to the given specifications. * Phase 4:   + Add and test a JavaScript solution for a predefined problem to the page.   Each phase will be version controlled and uploaded to the ScreenCraft Development Server.  Wireframes  The wireframes below illustrate the desired layout of the site. Your semantic HTM should match the layout (NOTE: Images are not required in Phase 1)   |  |  | | --- | --- | | Basic Layout The site will have a full width header and navigation, a main area that is approximately 960 pixels wide, and a full width footer.  A screenshot of a computer  Description automatically generated | Articles and Aside Concept Please note that the articles should have two columns, but they may not have even lengths. This is a decision to be made by the developer.  A screenshot of a web page  Description automatically generated | | Article and Aside Layout The article will have at least a header and section content. Footer may be omitted as required.  The section content may be used to split the article into subsections.  The aside is structured with multiple sections.  A screenshot of a computer  Description automatically generated | Image Layout options Images may have the caption over the top of the image at the bottom of the image space, or the image may occupy the top section of the area with its caption underneath the image.  A screenshot of a photo layout  Description automatically generated | | Aside Layout The aside has multiple sections. One of the sections will have your name contained in a footer as shown. Others may or may not have a footer, but still must contain a heading.  A screenshot of a computer  Description automatically generated |  | |
| Site Structure Requirements |
| All files must apply the naming conventions outlined in Appendix B: Naming Conventions.  Code must apply suitable code style as advices in Appendix A: Code Style Guidelines.  The project has a defined structure outlined in the Appendix C: Site Folder Structure.  The primary page in each phase must be named index.html.  Content details and additional steps will be made more explicit in each subsequent part of the portfolio.  Ensure you DO NOT complete more than one phase of development at any time. |
| General Instructions |
| Complete each step of the document in the order given.  All work must be completed at the CLI, except for starting the development environments.  When a step requires you to confer with the lecturer, ensure you do so, and make notes as required in spaces provided in this document. |
| Answering Questions |
| When a step includes a question, you must attempt to answer it.  There is a minimum and maximum number of words to use for each answer.  Any step that requires answers to be provided will have a space in this document immediately after each step. The answer space will expand with the content you type or images you paste.  Resize images to fit the space provided, ensuring they are still legible.  Do not include a whole screenshot, just the required details.  If a step has more than one question, these maxima and minima are a total for all the questions in that specific step.  All answers must be in complete sentences unless indicated.  *If required, make sure to add any code you’ve written in a separate file to your submission.*  *DO NOT put long pieces of code (over 10 lines) in this document.* |
| Sources of Information |
| In industry, it is good practice to keep track of where information was obtained. This is especially true if it is a written document, or even code.  If you answer any questions using information from web sites, please include the site name and URL (Web site address) after the answer.  Likewise, include the title and author for books and magazine articles.  For example:   * RS Electronics Ltd:  <https://au.rs-online.com/> * Slack API Documentation, Users List Method:  <https://api.slack.com/methods/users.list> |
| Code Storage |
| We advise that you create a private GIT repository on GitHub and use this to store a copy of your work.  This assessment has instructions on setting up version control and a remote (private) repository. Do not initialise any version control until told to do so.  To ensure you have a recent copy of your code you **must** use **AT LEAST ONE** of the following methods:   * Private Git repository, and/or * Cloud Storage (OneDrive within your college Office365), and/or * Keep a copy on a USB thumb drive.   Backing up to One Drive or to USB is best done by compressing the project folder up before copying to either OneDrive or USB. |
| Code Style |
| Please see **Appendix A: Code Style Guidelines** for details on the code style(s) to use for the assessment. |
| Professional Code and Documentation Requirements |
| This document contains appendices that cover the professional requirements when creating files, code, project structures and more.  These requirements must be adhered to.  No external frameworks are allowed in this part of the portfolio. This means that you may not employ frameworks such as Bootstrap, jQuery, Tailwind CSS, et al in your work. |
| Assessment Structure & Development Process |
| To complete this assessment, you will be asked to complete five phases.  The first four demonstrate the ability to:   * create semantic and structurally HTML suitable for a defined project, * add styling to implement a predefined layout, * add further styling and content to create a site that looks visually appealing, and * create a JavaScript based solution to a problem and have it integrate with HTML and CSS.   The final, fifth phase, requires you to implement a single page design using the TailwindCSS framework.. |
| The Development Phases  It is important that you do not start subsequent phases of the development process until the previous phase is completed and submitted.  The development phases include:   * Development Phases 0 & 1 (C-WT-AT2-POR-Phase-1):   + Setting up project for site.   + Create phase-1 folder and other required folders for the project.   + Download and add the provided files and folders to your project.   + Read scenario and this document.   + Seek clarification as required.   + Create suitable semantic markup for the provided design concept and content.   + Deploy, Validate and Test your code. * Development Phase 2 (C-WT-AT2-POR-Phase-2):   + Duplicate the Phase-1 files into a new folder.   + Work on the files in this new Phase-2 folder.   + Add a CSS Reset/Normaliser to the Web Page HTML in the correct location.   + Create a CSS file for the site and link it to the page after the CSS Reset/Normaliser.   + Add the required basic styling to create desired page layout.   + Deploy, Validate and Test your code. * Development Phase 3 (C-WT-AT2-POR-Phase-3):   + Duplicate Phase-2 files into a new folder.   + Work on the files in this new Phase-3 folder.   + Add the new content, imagery, and other media to the page.   + Complete the styling of the page.   + Deploy, Validate and Test your code. * Development Phase 4 (C-WT-AT2-POR-Phase-4):   + Duplicate the Phase-3 files into a new folder.   + Create JavaScript solution for provided problem.   + Display results in designated area on page.   + Deploy, Validate and Test your code. * Development Phase 5 (C-WT-AT2-POR-Phase-5):   + Implement TailwindCSS into the provided site index page.   + Provide one of the designated layouts for this page.   + Ensure all other phases are correctly linked to this index page.   + Test code works as expected.   All Phases require you to deploy the solution to a provided web host and to demonstrate the solution to your lecturer. You may be asked to explain your code. |

# Assessment Steps

Complete the steps in the order given.

| **STEP** | **Task to perform** |
| --- | --- |
| 00 | Complete the Front Page Make sure you have filled out the front page of this document.  Familiarise yourself with the content and document your progress in this assessment. This means, **READ** the **WHOLE** assessment **AT LEAST ONCE** before starting any work.  Make notes on the requirements of this assessment, as details appear as you progress through and are not given in one location.  This is very similar to how details emerge during the continuous development interviews and meetings with a client.  At any stage during this assignment, you may consult the stakeholder(s) or their representative(s).  Max Sentences: N/A |
|  | Intentionally left blank |
| 01 | Setting Up for Development We recommend that all work for this and following portfolios are kept in:   * C:\Users\XXXXXXXX\Source\Repos * ~/Source/Repos   We then recommend that you keep this cluster’s work in a WebTech folder.  To create this folder use:  PC (Using the Laragon terminal):   * cd %USERPROFILE% * mkdir –p Source\Repos\WebTech   Mac:   * cd ~ * mkdir –p Source/Repos/WebTech   Then use the command:   * cd Source/Repos/WebTech   All steps after this we will presume you will use this new folder as the starting point. Create Project in IDE Download the provided compressed file that contains the site folder structure and sample data.  Extract the folder to your storage location, and rename the folder to:   * C4Prog-WT-23S2-Portfolio-XXX   Where XXX is replaced by your initials.  For example: C4Prog-WT-23S2-Portfolio-AG. Update Git Ignore You are provided with a predefined “.gitignore” file to reduce some aspects of version control, especially with files going to the remote.  You will want to add the “.idea”, “.idea/\*” and “.idea/\*\*” patterns to the ignore file. Commit and Push Code Using any suitable method, you will now create, commit, and push to a private repository:   * Initialise the whole local project as a git repository. * Check all the content into the local repository. * Create a PRIVATE remote repository using any appropriate method.   Push this starter code to the remote private repository.  Max Words: N/A |
| A01 | What is the name and the URL of your private repository?  |  |  | | --- | --- | | Repository Name: | web\_tech | | URL: | <https://github.com/GooseTafe/web_tech> | |
| 02 | Create HTML for Page The company has provided you with a basic wireframe structure for the website which must be used as the base for the implementation.  The content of the site is provided and is downloadable from the assessment page (see Phase 0: setting up).  Once you have downloaded the content, use the Wireframes provided to determine your HTML structure for the page.  Implement structurally and semantically sound HTML in a new HTML page in the phase-1 folder in the provided sample site structure.  Add the new file(s) to your repository.  Commit and push to the private remote.  Please see Appendix YYY on how to create suitable commit messages. |
| 03 | Validating the HTML Use the WC3 Validation engine to verify your HTML is valid.  Copy the evidence for validity into the evidence folder inside phase-1.  Screenshots of the final product for this phase will also be included as evidence in the above folder.  Add the evidence folders to the “.gitignore” file in the repository.  Once completed check in all your code to your local repository and push it to your private remote. |
| A03 | Add a screenshot of the validation results for your code |
| 04 | Adding the Provided Content Use the content provided in the phase-1.txt file to fill out the HTML page.  Verify the page still passes the validation. |
| A04 | Add a screenshot of the validation results for your updated code |
| 05 | Testing the Web Page During the previous steps, you have created and updated a single web page.  During this it is possible that you have been checking what it looked like on at least one web browser.  You are now to verify that the site works on at least two browsers that use different rendering engines.  Take a screenshot of the page in the two browsers.  Paste these into the two sections below.  Make sure the images are no larger than 14cm and no smaller than 12cm wide. |
| A05.1 | Which browser is this screenshot from? Google chrome Screenshot |
| A05.2 | Which browser is this screenshot from? Firefox Screenshot |
| A05.3 | Which browser is this screenshot from? Microsoft Edge Screenshot |
| 06 | Commit and Push Code Using any suitable method, commit the completed code to version control and push to your private repository.  See Appendix XXX for details on writing good commit messages. |
| 07 | Demonstrate to Assessor Demonstrate to the assessor.  Obtain their feedback and prepare for phase 2 of the site development.  (Your lecturer may ask you to share your GitHub repository with them) |
| END | Submission of Portfolio Work To submit the portfolio, do the following:   * Save this document with your answers. * Compress the portfolio folder using 7-Zip * Open Blackboard, locate and open the AT2-PT1 Por Par 1 assessment * Upload this completed assessment document. * Upload the Word Document * Upload the compressed site files. * Click submit.  All answer documents MUST be submitted in Microsoft Office 365’s Word format. It is important that any PDF and Word documents are separate from the portfolio compressed folder as this makes it much easier for assessment. |

# Appendix A: Code Style Guidelines

The following guidelines should be applied to your code as it is developed.

Many may be applied via the use of PhpStorm, PyCharm or similar plugins and code formatting.

### PHP Code (General)

Please refer to the PHP PSRs:

* <https://www.php-fig.org/psr/psr-1/>
* <https://www.php-fig.org/psr/psr-12/>
* <https://www.php-fig.org/psr/psr-4/>

### Applications Built with Laravel

Please refer to the following articles:

* <https://dev.to/lathindu1/laravel-best-practice-coding-standards-part-01-304l>
* <https://dev.to/lathindu1/laravel-best-practice-coding-standards-part-02-a40>

### HTML Code

Please refer to the Google Style guide:

* <https://google.github.io/styleguide/htmlcssguide.html>

Note that PhpStorm and WebStorm will format code to their own standard, which is acceptable.

### JavaScript

Refer to the Google JS Style guide:

* <https://google.github.io/styleguide/jsguide.html>

Note that PhpStorm and WebStorm will format code to their own standard, which is acceptable.

### Python Code

Your code will follow the PEP 8 standard.

PyCharm will format code very close to the PEP-8 standard.

### JSON Code

JSON should be formatted in an appropriate manner.

*Readability Counts   
- Zen of Python*

# Appendix B: Naming Conventions

## HTML, CSS and JS Files and folders

* No spaces.
* Alpha-numeric characters only except minus/hyphen (-) and full stop (.).
* All lowercase letters.
* Use hyphens (dash) between words.

## File extensions

The following file extensions must be used:

|  |  |
| --- | --- |
| HTML pages | .html |
| JS | .js |
| CSS | .css |
| PHP | .php |
| Node | .js |
| React | .jsx |
| Python | .py |
| JSON | .json |
| XML | .xml |

# Appendix C: Site Folder Structure

The diagram below outlines a folder structure for the portfolio from Phase 1 to Phase 4:

# Appendix D: Conventional Commit Messages

Conventional Commit Messages, or Semantic Commit Messages, are a way to improve yur commit messages and how you document what work has been completed.

A small change to your behaviour when writing commit messages will make a HUGE difference to a team, as well as yourself.

This brief outline is based on the following references:

* <https://www.conventionalcommits.org/en/v1.0.0/>
* <https://seesparkbox.com/foundry/semantic_commit_messages>
* <http://karma-runner.github.io/1.0/dev/git-commit-msg.html>

All links were retrieved on Friday 4th August 2023.

## How to Create SCMs/CCMs

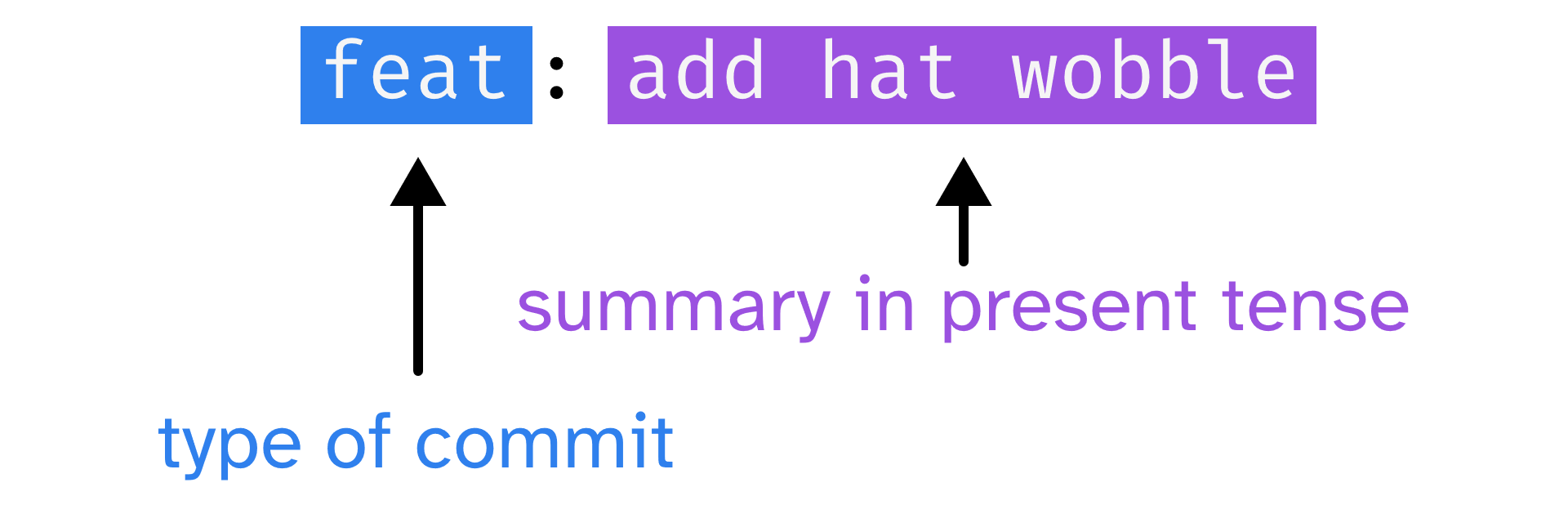
See how a minor change to your commit message style can make you a better programmer.

The format of a SCM is:

<type>(<scope>): <subject>

Where the <scope> is optional

### Example



## Types of Commit

* feat: (new feature for the user, not a new feature for build script)
* fix: (bug fix for the user, not a fix to a build script)
* docs: (changes to the documentation)
* style: (formatting, missing semi-colons, etc; no production code change)
* refactor: (refactoring production code, e.g. renaming a variable)
* test: (adding missing tests, refactoring tests; no production code change)
* chore: (updating grunt tasks etc; no production code change)

## Types of Scope

The scope types pertain to components/features of the application/code.

For example:

docs (users): create user management documentation (add, edit)

Here are some possible scopes that may be used.

* init: initialisation file/code
* runner
* watcher
* config: configuration files/code
* web-server: code pertaining to web-server configuration, execution and so on
* proxy: proxy configuration, execution, and such