

FRONT-END DEVELOPMENT

Congratulations, you have made it to the next phase of DTT's application procedure!

This phase will consist of a development test made with Vue.js and JavaScript. During your time at DTT, you will work for strong international brands, including Greenpeace, Randstad, ING, Philips, Rabobank, FOX Sports, and KPMG. Through this test, you can demonstrate that you are capable of contributing to DTT and her clients. DTT promises her clients a detail orientated approach, a high quality standard for deliveries, and a work ethic that goes that extra mile: we look for applicants that understand these values.

This test gives us a better understanding of how you work, and provides you with insight into what you can expect when working at DTT.

Keep in mind:

- I. Log the hours you spend on the project in the provided Excel document, provide a detailed description of your activities to help us in the grading process.
- II. We strongly advise you to complete the **user stories** within 30 hours, and to adjust your scope based on this advisory deadline. Time taken for **bonus points** will not be included in the final time evaluation.

THE TEST

We challenge you to create a web application for house listings, which allows the user to view a list of houses that are currently available for sale, and create, edit and delete their own listings.

You can find the design of the web app in the following links. The designs contain some extra elements, which are part of the Bonus points.

- i. [Desktop workflow](#)
- ii. [Mobile workflow](#)

API information

The data will be fetched from a remote source, using our existing API. Please, note that the API requires an access key **header** to obtain, create, edit or delete houses.

- i. You can get an API key from our [Signup page](#).
- ii. fill your first and last name and your email address.
- iii. Make sure you **save your API key**. Once you leave the page, you will not be able to access your API key.
- iv. Note that the images are **only accessible** from localhost:8080.

The API has detailed documentation, which can be found [here](#).

User stories

DTT uses user stories as a baseline for all our (technical/design) documentation. Make sure your test adheres to the user stories and conditions of satisfaction as stated below.

For the minimal viable product, the following user stories should be implemented:

User story 1: As a user, I want to be able to see a fixed navigation header, so that I can easily navigate through the app.

- *I want to be able to navigate to the 'Houses' page.*
- *I want to be able to see what page is currently active.*

User story 2: As a user, I want to be able to see an overview of all available houses within the homepage of the web app, so that I can see what the app has to offer.

- *I want to be able to see the address, postal code and city of the listing.*
- *I want to be able to see the image, size, the number of bedrooms and bathrooms of the listing.*

User story 3: As a user, I want to be able to search through the available houses by at least one of the house listing properties, so that I can find specific houses based on search criteria.

- *I want to be able to clear my search input, after I start typing.*
- *I want to be able to see a result indication, informing me about the number of results that are found.*
- *I want to be able to see if my search input did not match any results.*

User story 4: As a user, I want to be able to sort the available houses shown on the 'Homepage' by at least one option (price or size), so that I can sort based on my preference.

User story 5: As a user, I want to be able to access the details of a house, so that I can read more in-depth information about my selected house.

- *I want to be able to see the address, postal code and city of the listing.*
- *I want to be able to see the description, image, number of bedrooms and bathrooms, the size, construction and if the listing has a garage or not.*
- *If it is my own listing, I want to be able to delete and edit the listing.*

User story 6: As a user, I want to be able to create my listing(s), by completing all the required fields in a form.

- *I want to be able to add the address (street name, House number, Addition, Postal Code and City) of my listing.*
- *I want to be able to set an image, a description, a price, the size, the amount of bedrooms and bathrooms, the construction date whether my listing has a garage or not.*
- *If I have not completed all the required fields or there is a validation error in the form, I should see an appropriate error message.*
- *After completing the form and creating the listing, I should be redirected to the newly created listing detail page.*

User story 7: As the owner of a listing, I want to be able to have an edit and delete button on the detail page of a house that I own, so that I can navigate to the edit page of my listing or delete the listing.

User story 8: As the owner of a listing, I want to be able to edit my listing(s).

- *I want to be able to edit the address (street name, House number, Addition, Postal Code and City) of my listing.*
- *I want to be able to edit the image, description, price, size, amount of bedrooms and bathrooms, the construction date and whether my listing has a garage or not.*
- *If I have not completed all the required fields or there is a validation error in the form, I want to be informed with an appropriate error message.*

User story 9: As the owner of a listing, I want to be able to delete my listing(s).

- *I want to see a warning before the successful deletion of my listing.*

Additional requirements:

- i. The project is made in JavaScript using Vue.js as a framework.
- ii. General coding standards and principles are followed ([HTML](#), [CSS](#)).
- iii. UI libraries are not allowed (e.g. Bootstrap, Vuetify, etc).
- iv. Code is adequately commented and consistently formatted.
- v. Vuex is used.
- vi. Hours are tracked and logged.

Bonus points for a good impression (all optional)

As a Front-end intern, you are expected to have a solid understanding of Vue.js and JavaScript and are able to make use of APIs in a documented, readable, well-structured manner, and produce scalable code. We are more likely to invite candidates for a follow-up interview when we can accurately gauge one or more of these aspects. As such, the bonus points are very 'free format', allowing you to decide which impression you want to leave us with.

A few ideas for extra points:

- i. Give the ability to the user to like a house and add it to her/his list of favorites.
- ii. Extend the user experience by allowing the user to see the history of the houses he/she has viewed.
- iii. Expand the functionality of the website with a new page, in which the user can see all of her/his listings.
- iv. Allow the user to see recommendations on the detail page based on one or more criteria.
- v. Add an "About" page so the user can read some information about the web app and the developer/DTT/something else.
- vi. Add additional sorting method(s) to the website so that the user has more variety.
- vii. Add filter option(s) so that the user can specify in more detail what he/she is looking for..
- viii. Have a fun idea? We'd love to see it! :)

These points are completely optional and do not count towards your 'total time'. They must adhere to the requirements of the test. We do not negatively judge a test that does not include bonus features, but differentiating your test does increase your chances to get invited for a follow-up.

Delivery

When you are satisfied with your web app, **zip** your project (including your hour log and API key) and send a [WeTransfer](#) link to your person of contact at DTT.

If you worked with a GIT repository, please follow the steps below:

- Invite *apply@d-tt.nl* as a contributor with Admin privileges
- Make sure that the repository includes the full project source code
- Make sure that the repository includes a copy of any other deliverables e.g.:
 - your completed hours log.
 - your API key.
 - etc.

Our review

We review your test on different criteria. Functional requirements are graded on their completeness and the chosen method of implementation. The quality and structure of code are graded on (among other factors) consistency, reusability and extensibility. The design requirements are graded on the accuracy. The hour log is graded on its completeness, language, and level of detail.

Among other factors, the following points are essential to us:

- **Consistency:** naming, structure, format. There is consistent naming, structure and formatting throughout the code.
- **Code and component (Vue.js) reusability and structure:** The code is reusable and maintainable. There is no unnecessarily repeated and duplicated code. The Vue.js components are reusable and well-structured.
- **Accuracy of the design implementation and responsiveness:** The design has been implemented accurately and the web app is fully responsive.
- **Documented code:** Comments are punctuated, straight to the point, grammatically correct and used consistently to make the code readable.

Adherence to these criteria indicates an approach to work we greatly value.

Questions?

If you have any question regarding the requirements of the assignment, or if you are stuck on a problem for too long, please don't hesitate to contact us.

More DTT

Feel free to have a look at all our apps at: <https://www.d-tt.nl/en/apps>.