♔ **GOV.UK**

Service Manual

beta Complete our quick 5-question survey to help us improve our content (https://www.smartsurvey.co.uk/s/HGLV5/?c=/service-manual/helping-people-to-use-your-service/testing-for-accessibility).

1. Service manual (https://www.gov.uk/service-manual)
2. Accessibility and assisted digital (https://www.gov.uk/service-manual/helping-people-to-use-your-service)

Accessibility and assisted digital

Testing for accessibility

Give feedback about this page (https://www.gov.uk/contact/govuk)

From:
Accessibility community (https://www.gov.uk/service-manual/communities/accessibility-community)
Last updated:
almost 2 years ago

Page contents:

You need to make sure your service meets level AA of the Web Content Accessibility Guidelines 2.1 (https://www.gov.uk/service-manual/helping-people-to-use-your-service/understanding-wcag-20) (WCAG 2.1) as a minimum.

If the service doesn’t meet WCAG 2.1 AA, you may be breaking the law.

If you’re working to meet the Service Standard (https://www.gov.uk/service-manual/service-standard), you’ll also need to:

- make sure the service works with the most common assistive technologies (https://www.gov.uk/service-manual/technology/testing-with-assistive-technologies) - screen readers or speech recognition software, for example
- test the service with disabled users (https://www.gov.uk/service-manual/user-research/running-research-sessions-with-people-with-disabilities) and with older users

The best way to meet accessibility requirements is to:

- think about accessibility from the start

- run your own accessibility tests regularly throughout development
- get a formal accessibility audit before you go into public beta

Think about accessibility from the start

Consider accessibility at every stage. You might find the Government Digital Service's (GDS) accessibility user profiles (https://www.gov.uk/government/publications/understanding-disabilities-and-impairments-user-profiles) useful.

Start thinking about technical accessibility from alpha. When you're discussing ideas and developing concepts, consider:

- whether what you're thinking about meets the WCAG design principles (https://www.gov.uk/service-manual/helping-people-to-use-your-service/understanding-wcag-20#wcag-20-design-principles)
- how people with impairments to their sight, hearing, movement, memory or thinking might use it

You should run regular tests as soon as you start writing production code. Once your service moves into public beta, run tests every time you add a new feature.

When you're working on rough prototypes, you don't need to worry about your code being accessible. But it's useful to check that things like the colour contrasts you're using are accessible.

Working in this way helps you identify and fix issues early. It's much more expensive to unpick and fix them at a later stage.

Testing your code

Test your code regularly, using both automated testing and manual testing. These tests will also uncover issues with design and content.

It's important to do both types of testing - you'll miss some issues if you only do automated testing.

Manual testing

Use the accessibility checklist created by 18F (https://accessibility.18f.gov/checklist/) (the US government's digital agency) to help you test for common accessibility problems, including:

- lack of keyboard accessibility (important because some have a disability which means they rely on using a keyboard to navigate websites)
- link text that's not descriptive (for example,'click here' links)
- lack of colour contrast for text and important graphics and controls
- images not having meaningful alt text (where alt text is needed)
- online forms (https://design-system.service.gov.uk/components/) not being marked up correctly, so the right control is associated with the right label

Some browsers have tools that make it easier to find accessibility problems in the Document Object Model (DOM). For example, accessibility Inspector for Mozilla Firefox (https://developer.mozilla.org/en-US/docs/Tools/Accessibility_inspector) and the accessibility features in Chrome DevTools (https://developers.google.com/web/tools/chrome-devtools/accessibility/reference).

Automated testing

For automated testing GDS uses a range of tools, including:

- aXe (https://www.deque.com/axe/)
- Wave (http://wave.webaim.org/extension/)
- Tenon (https://tenon.io/)
- SiteImprove (https://chrome.google.com/webstore/detail/siteimprove-accessibility/efcfolpjihicnikpmhnmphjhhpiclljc)

See the GDS accessibility team's audit of the most used accessibility tools (https://alphagov.github.io/accessibility-tool-audit/) if you're not sure which tools to use.

Testing with assistive technologies

Your service needs to work on the most common browsers and devices (https://www.gov.uk/service-manual/technology/designing-for-different-browsers-and-devices) people will use to access your service.

You must also make sure your service works with common assistive technologies (https://www.gov.uk/service-manual/technology/testing-with-assistive-technologies).

This means:

- where possible, doing some testing with assistive technology yourself (you can ask to use the GDS empathy lab (https://docs.google.com/forms/d/e/1FAIpQLSdMU-SXWtBLcq9y3fdMXPNDHKIVwlcnXwveUEtJpgYw9zEOGg/viewform) to help your team learn about assistive technology)
- finding user research participants (https://www.gov.uk/service-manual/user-research/find-user-research-participants) who use assistive technology
- asking for assistive technology testing to be included in your accessiblity audit

Getting an accessibility audit

You must get an accessibility audit (https://www.gov.uk/service-manual/helping-people-to-use-your-service/getting-an-accessibility-audit) - and fix any issues - before your service moves into public beta.

Useful resources

You can start some simple tests with:

- W3C Easy Checks (https://www.w3.org/WAI/test-evaluate/preliminary/) - a few quick things to help you start to assess how accessible a web page is
- Basic screen reader commands for accessibility testing (https://developer.paciellogroup.com/blog/2015/01/basic-screen-reader-commands-for-accessibility-testing/), from the Paciello Group
- WCAG report generator (https://www.w3.org/WAI/eval/report-tool/#/)

You can also read some blog posts on:

- dos and don'ts on designing for accessibility (https://accessibility.blog.gov.uk/2016/09/02/dos-and-donts-on-designing-for-accessibility/)

- how the Government Digital Service did accessibility testing on GOV.UK (https://gds.blog.gov.uk/2012/01/20/user-testing-accessibility/)

Related guides

You may also find these guides useful:

- Making your service accessible: an introduction (https://www.gov.uk/service-manual/helping-people-to-use-your-service/making-your-service-accessible-an-introduction)
- Using, adapting and creating patterns (https://www.gov.uk/service-manual/design/using-adapting-and-creating-patterns)
- Using progressive enhancement (https://www.gov.uk/service-manual/technology/using-progressive-enhancement)

From:
Accessibility community (https://www.gov.uk/service-manual/communities/accessibility-community)
Last updated:
almost 2 years

Last update:
6 March 2019

Guidance first published