

Support opening DevTools via CDP command

Attention: Externally visible, non-confidential

Author: Nikolay Vitkov

Status: Accepted ▾

Created: 2025-07-10 / **Last Updated:** 2025-07-16

One-page overview

Summary

Introduce a new CDP command that opens DevTools for a given target . This opens DevTools on the remote machine similar to clicking Inspect on the page.

The use case is in the E2E testing for DevTools where we want to open DevTools on demand for a specific page.

Platforms

Desktop

Value proposition

This change removes the need of a flag setting on the `--auto-open-devtools-for-tabs` for any tooling that would want to open DevTools. Allowing clients to decide when they need DevTools, removing the need to close DevTools targets that are not wanted when the above flag is on. This also gives them the TargetId of the newly opened DevTools which removes the guesswork when using the flag.

Code affected

Chrome DevTools Protocol

Signed off by

Name	Approval status
Danil Somsikov	LGTM ▾

Alex Rudenko	LGTM ▾ with comments
Mathias Bynens	LGTM ▾
Alex N. Jose	LGTM ▾
Maksim Sadym	LGTM ▾
⊞ Person	Review not started ▾

Background

In Chrome there are multiple ways of opening DevTools from the UI (inspect, shortcuts, menu). There is only a single way to open DevTools programmatically by using the Chrome flag `--auto-open-devtools-for-tabs`, which as the name implies opens DevTools for each new tab created.

Design

We propose adding a new command on the Target domain called `Target.openDevTools`

```
domain Target
  experimental command openDevTools
  parameters
    TargetID targetId
  returns
    TargetID targetId
```

The command will take a `targetId` and attempt to open DevTools. The implementation will try to reuse the `DevToolsWindow::OpenDevToolsWindow()`. And return the information for the created DevTools target. Additionally we could try to see if we can deduce the current TargetId by session while preserving the security requirements. That will allow us not to send the command on the BrowserSession.

For headless-shell do not include it in `headless/protocol_config.json` which will result in `MethodNotFound`.

Alternatives considered

Add a new parameter to the `Target.createTarget` named `devTools: boolean` which will create a new DevTools target for the given Session. Cons of this approach is that all other parameters are inapplicable to this DevTools Target and should be ignored including the mandatory `url`. Extending it with other functionality (follow-ups) will further add complexity to the command.

Rollout plan

Waterfall

Core principle considerations

Speed

N/A

Security

The ability to open DevTools may have security risks. We will mitigate this risk by only exposing it to trusted clients. This will be similar to the `--auto-open-devtools-for-tabs` that requires the browser to be started in a specific way.

We will use the same mechanism used for `Target.exposeDevToolsProtocol` utilizing the `AccessMode::kBrowser` flag to allow only externally connected sessions to send the command.

We will not expose this functionality to untrusted clients such as DevTools and Extensions.

The implementation will also respect the already existing rule of when DevTools is available by utilizing `IsInspectionAllowed`.

Simplicity

N/A

Accessibility

N/A, no UI changes

Testing plan

Browser tests

Because we need to create the actual DevTools target, we can't use any test relying on the content shell which does not implement DevTools with inspector-protocol test being one such suite.

The plan is to put the test inside

chrome/browser/devtools/protocol/devtools_protocol_browsertest.cc

Follow-up work

Update the [enum class DevToolsOpenedByAction](#) to specify a new way DevTools was opened. This will show up in the corresponding logs.

Add an additional parameter to the newly added command to specify which mode DevTools is open (undocked, docked left, docked right, docked bottom). This can help people control how DevTools is displayed, reducing a step of opening and then changing the mode.