Making the Flutter CLI work with custom or third-party embedders

The experimental "custom device" feature of the Flutter CLI allows you to make custom or third-party embedders such as flutter-pi or the sony embedder work with the Flutter CLI and use features such as hot-reload, hot-restart, debugging and DevTools. Profile/release mode and plugins are not supported right now.

Currently, `custom-devices` support is only available in the master channel.

Enabling custom-devices for your Flutter SDK

Enabling the feature is as simple as `flutter config --enable-custom-devices`.

The custom devices config file

After that, a config file will have been created. Its path should be printed out when you run `flutter custom-devices`.

The config file contains all the configured custom devices. To see documentation for all the config options, open the file in VS Code. The file has a JSON schema so in any editor supporting JSON schemas (currently only VS Code) you'll get autocompletion, documentation, examples and so on.

Configuring your custom device

If your device is reachable via ssh, you can add it to the config with a guided setup using `flutter custom-devices add`.

The setup really is self-explanatory, but for example for my Raspberry Pi, the setup looks like this:

```
hannes@pop-os:~/devel$ flutter custom-devices add
Please enter the id you want to device to have. Must contain only alphanumeric
or underscore characters. (example: pi)
pi
Please enter the label of the device, which is a slightly more verbose name for
the device. (example: Raspberry Pi)
Raspberry Pi
SDK name and version (example: Raspberry Pi 4 Model B+)
Raspberry Pi 4 Model B+
Should the device be enabled? [Y/n] (empty for default)

Please enter the hostname or IPv4/v6 address of the device. (example:
raspberrypi)
hpi4
Please enter the username used for ssh-ing into the remote device. (example: pi,
empty for no username)
```

```
Please enter the command executed on the remote device for starting the app.
"/tmp/${appName}" is the path to the asset bundle. (example: flutter-pi
/tmp/${appName})
flutter-pi /tmp/${appName}
Should the device use port forwarding? Using port forwarding is the default
because it works in all cases, however if your remote device has a static IP
address and you have a way of specifying the "--observatory-host=<ip>" engine
option, you might prefer not using port forwarding. [Y/n] (empty for default)

Enter the command executed on the remote device for taking a screenshot.
(example: fbgrab /tmp/screenshot.png && cat /tmp/screenshot.png | base64 | tr -d
' \n\t', empty for no screenshotting support)

Would you like to add the custom device to the config now? [Y/n] (empty for
default)

Successfully added custom device to config file at
"/home/hannes/.config/flutter/custom_devices.json".
```

If your device is a little more complicated to setup, for example if your device is the one the flutter SDK is running on, you can configure it by editing the config file directly. (As mentioned above, I heavily recommend VS Code for that)

That's it

After that, the device is configured and if you run `flutter devices`, it should show your custom device there (if it is reachable).

You should be able to run apps on your device using `flutter run -d devicename` or using your IDE.