

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with subtle diagonal lines.

Releases: Part 2

The Application



Previously

- /lib contains our compiled applications / dependencies
- /releases contains specific release version information
- /erts... includes what we need to boot the BEAM
- /bin contains various scripts Distillery creates for us



Next:

1. The Basics

2. The Application [you are here]

- A. Learn about each file type (.app, .rel, .boot)
- B. Investigate how priv directories work
- C. Learn about run_ert/do_ert and the IO pipe
- D. Upgrade our API to version 2 serving a file from a priv directory

3. The Environment

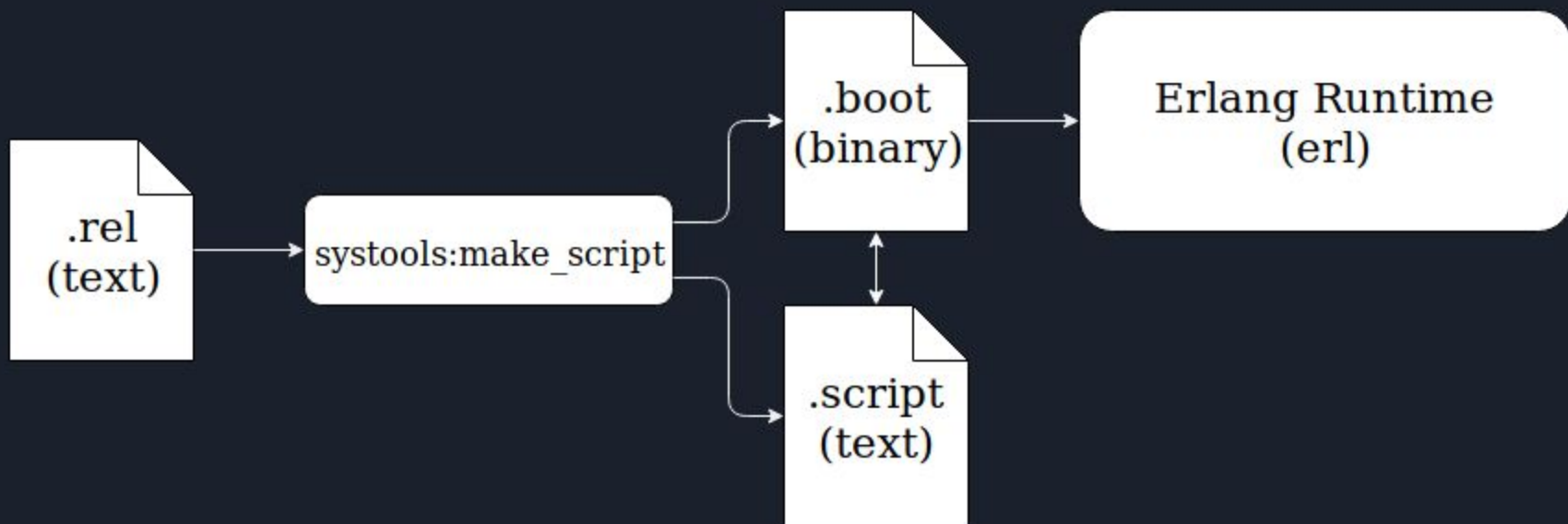
4. Distillery



The Boot Script

./releases/0.1.0/simple_api.script

```
{script, {Name, Vsn},  
 [  
   {progress, loading},  
   {preLoaded, [Mod1, Mod2, ...]},  
   {path, [Dir1, "$ROOT/Dir", ...]},  
   {primLoad, [Mod1, Mod2, ...]},  
   ...  
   {kernel_load_completed},  
   {progress, loaded},  
   {kernelProcess, Name, {Mod, Func, Args}},  
   ...  
   {apply, {Mod, Func, Args}},  
   ...  
   {progress, started}}].
```





The Release Resource File (.rel)

./releases/0.1.0/simple_api.rel

```
{release, {"simple_api", "0.1.0"},  
  {erts, "9.0"},  
  [{kernel, "5.3"},  
   {stdlib, "3.4"},  
   {poison, "3.1.0"},  
   {mime, "1.2.0"},  
   ...  
   {compiler, "7.1"},  
   {elixir, "1.5.2"},  
   {simple_api, "0.1.0"},  
   {iex, "1.5.2"},  
   {sasl, "3.0.4"},  
   {runtime_tools, "1.12"}}].
```



The Application Resource File:

`./lib/simple_api-0.1.0/ebin/simple_api.app`

```
{application, simple_api,  
  [{applications, [kernel, stdlib, elixir, logger, cowboy, plug, poison]},  
  {description, "simple_api"},  
  {modules, ['Elixir.SimpleApi', 'Elixir.SimpleApi.Application',  
             'Elixir.SimpleApi.Counter',  
             'Elixir.SimpleApi.Endpoint']},  
  {registered, []},  
  {vsn, "0.1.0"},  
  {extra_applications, [logger, cowboy, plug]},  
  {mod, {'Elixir.SimpleApi.Application', []}}}].
```



Mix file

Generate an app file with ``mix compile.app``

```
def project do
  [
    app: :simple_api,
    version: "0.1.0",
    elixir: "~> 1.5",
    start_permanent: Mix.env == :prod,
    deps: deps(),
  ]
end

def application do
  [
    extra_applications: [:logger, :cowboy, :plug],
    mod: {SimpleApi.Application, []}
  ]
end
```




Recap: To start an application we need...

- An app file describing dependencies for an app
- Beam files in proper directories ``lib/app-#{vsn}/ebin``
- A rel file describing which app files we depend on
- A script/boot file generated by ``:systools.make_script``



A note about priv directories Phoenix and assets.

- Each app has its own priv directory: `./lib/app-#{vsn}/priv`
- Each app's priv directory can be accessed within the app code with `:code.priv_dir``
- This is where your Phoenix assets end up, how they are served from Plug.



Erl pipes

- `run_erl <pipe_dir> <log_dir> "exec erl ..."`
- `to_erl <pipe_dir>`



Objectives:

- Checkout branch `part_2`
- Generate a new app file and compiled .beam files for the new feature
- Install the new files and get the VM to boot serving the example erl file in the priv directory.
- Attach a console to the running app