Phoenix: Plug.Conn cheatsheet

Request

```
conn.host          # → "example.com"
conn.method        # → "GET"
conn.path_info     # → ["posts", "1"]
conn.request_path  # → "/posts/1"
conn.query_string  # → "utm_source=twitter"
conn.port          # → 80
conn.scheme        # → :http
conn.peer          # → { {127, 0, 0, 1}, 12345 }
conn.remote_ip     # → { 151, 236, 219, 228 }
conn.req_headers   # → [{"content-type", "text/plain"}]
```

```
conn |> get_req_header("content-type")
# → ["text/plain"]
```

Response

Response

```
conn.resp_body     # → "..."
conn.resp_charset  # → "utf-8"
conn.resp_cookies  # → ...
conn.resp_headers  # → ...
conn.status        # → ...
```

Updating conn

```
conn
|> put_req_header("accept", "application/json")
```

Usually only useful for tests.

Sending responses

```
# Plug.Conn
conn
|> html("<html><head>...")
|> json(%{ message: "Hello" })
|> text("Hello")
```

```
|> redirect(to: "/foo")
|> redirect(external: "http://www.google.com/")
|> halt()
```

```
|> put_resp_content_type("text/plain")
|> put_resp_cookie("abc", "def")
|> put_resp_header("X-Delivered-By", "myapp")
|> put_status(202)
|> put_status(:not_found)
```

```
|> put_private(:plug_foo, "...")  # reserved for libr
```

```
|> send_resp(201, "")
```

Phoenix views

```
# Phoenix.Controller
conn
|> render("index.html")
|> render("index.html", hello: "world")
|> render(MyApp.ErrorView, "404.html")
```

```
|> put_layout(:foo)
|> put_layout(false)
|> put_view(ErrorView)
```

```
|> put_secure_browser_headers()
# prevent clickjacking, nosniff, and xss protection
# x-frame-options, x-content-type-options, x-xss-protection
```

```
|> put_new_view(ErrorView)  # if not set yet
|> put_new_layout(:foo)
```

```
layout(conn)
```

Other features

Other fields

```
conn.assigns          # storage of crap
conn.owner            # process
conn.halted           # if pipeline was halted
conn.secret_key_base  # ...
conn.state            # :unset, :set, :file, :sent, :chunked
```

Accepts

```
plug :accepts, ["html", "json"]
conn |> accepts(["html", "json"])
get_format(conn)  # → "html"
conn.accepts
```

Session

```
conn = fetch_session(conn)   # or plug :fetch_session

conn = put_session(conn, :message, "new stuff we just set in the
get_session(conn, :message)
conn = clear_session(conn)
```

```
conn
|> put_flash(:info, "Success")
|> put_flash(:error, "Oh no")
```

Also available: `flash` `cookie` `params`

Assigns

```
conn.assigns[:hello]
conn |> assign(:user_id, 100)
```

```
conn = async_assign(conn, :location, fn -> geoip_look
await_assign(conn, :location)
```