Debug_Protocol commands/modules

The following debug_protocol commands defined in `ocaml_dap` are modules that we need to register handlers for.

The `ocaml_dap` documentation can be found here.

The debug adapter protocol (defining what `ocaml_dap` implements) has documentation and specifications here

1. Breakpoints

- Set_breakpoints_command
- Breakpoint_locations_command (maybe)

2. Stopping/pausing execution

- Pause_Command
  - recieve this request, send a response, and then send a Stopped_event

3. Continuing/resuming execution

- Continue_command
  - recieve this request, send response, DO NOT need to send a Continued_event

4. Step in/out/forward

- Next_command
  - recieve this request, send a response, process the 'step', send a Stopped_event
- Step_in_targets_command
  - retrieves possible targets to use in stepIn command (optional I believe)
- Step_in_command
  - recieve this request, send a response, process the step (with adjusted granularity?), then send a Stopped_event
- Step_out_command
  - recieve this request, send a response, process the step, then send a Stopped_event

5. Call stack

- Stack_trace_command

6. Variable inspection

- Scopes_command
- Variables_command
- Evaluate_command

7. Restart

- Restart_command

  - either support it or emulate it

8. Disconnect/terminate/exit

- Disconnect_command
- Terminate_command
  - Terminate_event must be related somehow
- Exited_event
  - also probably related somehow

9. Initialization

- Initialize_command
  - Initialized_event
- Configuration_done_command
- Capabilities_event
  - has to do with capabilities changing (so maybe we will not support it)
- Launch_command
- Attach_command
  - We will probably emulate this by just launching when we respond to an attach command
- Threads_command