DOCUMENTING GSTREAMER

Centricular

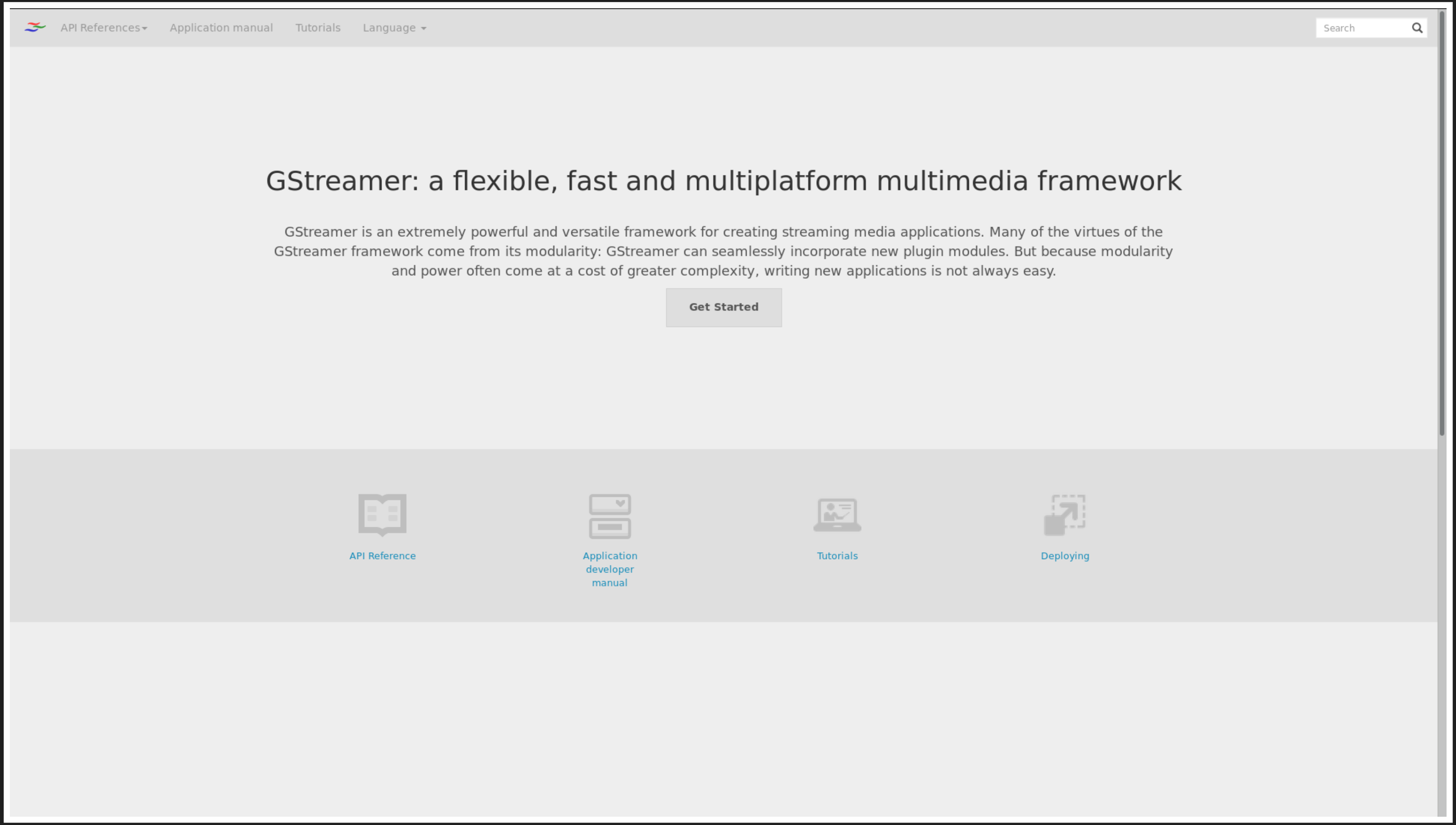
PORTING

- To a single format, `markdown` (slightly extended)
- To a single tool, `hotdoc`
- To a single* repository, `gst-docs`

Centricular

UNIFYING

- Previously a set of loosely-connected HTML documents
- Soon a unified developer portal



GStreamer: a flexible, fast and multiplatform multimedia framework

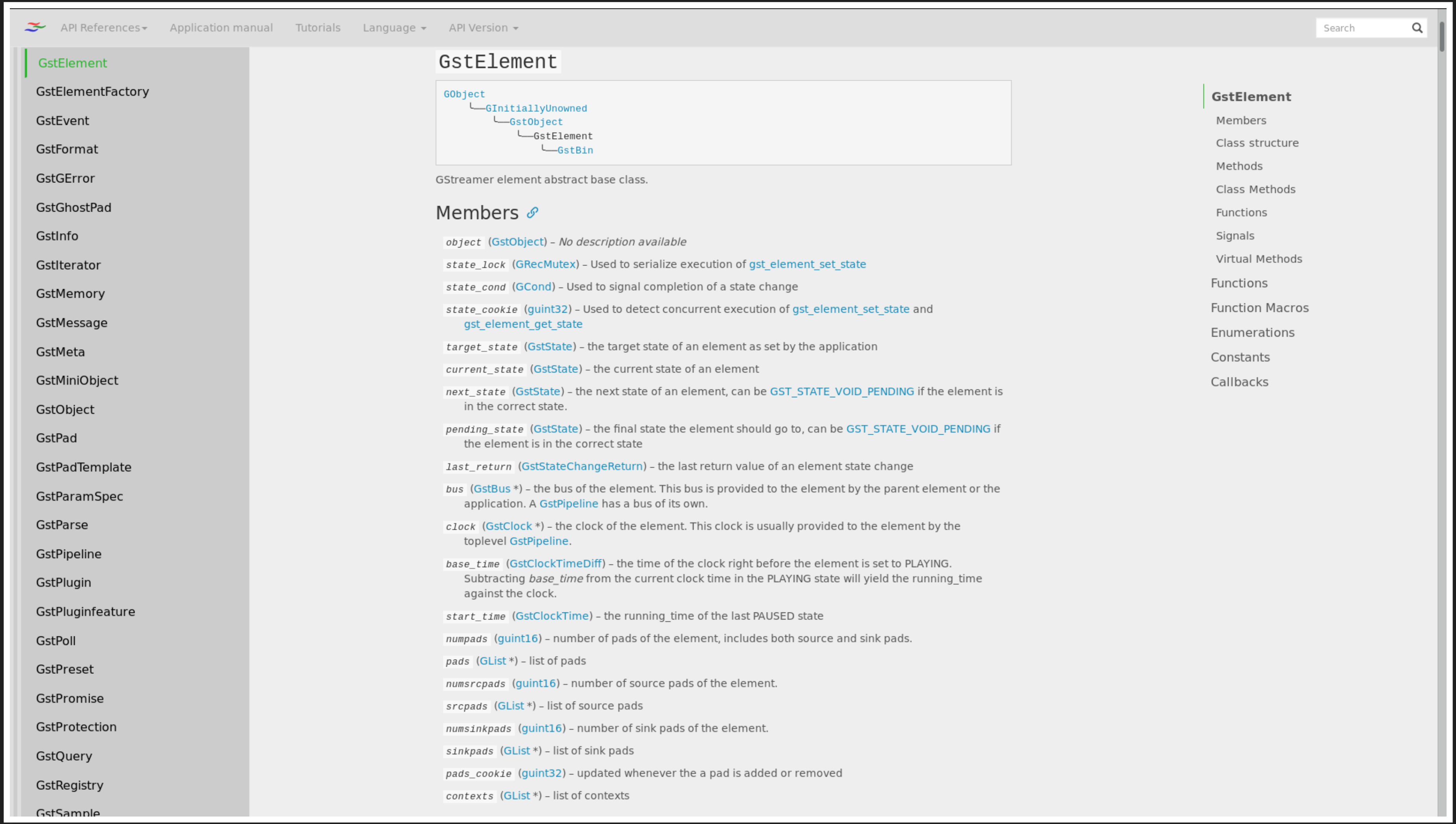
GStreamer is an extremely powerful and versatile framework for creating streaming media applications. Many of the virtues of the GStreamer framework come from its modularity: GStreamer can seamlessly incorporate new plugin modules. But because modularity and power often come at a cost of greater complexity, writing new applications is not always easy.

Centricular

DEVELOPER PORTAL

- Unified sitemap
- Unified search
- Unified theme

Centricular



GstElement

```
GObject
    ╰──GInitiallyUnowned
        ╰──GstObject
            ╰──GstElement
                ╰──GstBin
```

GStreamer element abstract base class.

Members

- `object` (GstObject) – *No description available*
- `state_lock` (GRecMutex) – Used to serialize execution of gst_element_set_state
- `state_cond` (GCond) – Used to signal completion of a state change
- `state_cookie` (guint32) – Used to detect concurrent execution of gst_element_set_state and gst_element_get_state
- `target_state` (GstState) – the target state of an element as set by the application
- `current_state` (GstState) – the current state of an element
- `next_state` (GstState) – the next state of an element, can be GST_STATE_VOID_PENDING if the element is in the correct state.
- `pending_state` (GstState) – the final state the element should go to, can be GST_STATE_VOID_PENDING if the element is in the correct state
- `last_return` (GstStateChangeReturn) – the last return value of an element state change
- `bus` (GstBus *) – the bus of the element. This bus is provided to the element by the parent element or the application. A GstPipeline has a bus of its own.
- `clock` (GstClock *) – the clock of the element. This clock is usually provided to the element by the toplevel GstPipeline.
- `base_time` (GstClockTimeDiff) – the time of the clock right before the element is set to PLAYING. Subtracting *base_time* from the current clock time in the PLAYING state will yield the running_time against the clock.
- `start_time` (GstClockTime) – the running_time of the last PAUSED state
- `numpads` (guint16) – number of pads of the element, includes both source and sink pads.
- `pads` (GList *) – list of pads
- `numsrcpads` (guint16) – number of source pads of the element.
- `srcpads` (GList *) – list of source pads
- `numsinkpads` (guint16) – number of sink pads of the element.
- `sinkpads` (GList *) – list of sink pads
- `pads_cookie` (guint32) – updated whenever the a pad is added or removed
- `contexts` (GList *) – list of contexts

Centricular

MARKDOWN

- Simple yet powerful language
- Low barrier to contribution
- Standardized (commonmark)
- Extensible*

HOTDOC

- "Static" output (no dedicated server needed)
- Powerful search
- Composability
- (Reasonably) fast
- Programming language-aware (C / Python / Javascript)

BUILD SYSTEMS

- hotdoc module in meson
- autotools snippets in hotdoc (vestigial, not recommended)

Centricular

GST-DOCS

- Home for standalone documentation
- Top-level node of the sitemap
- Leverages meson and hotdoc subprojects

PLUGIN DOCUMENTATION

```
adder = library('gstadder',
  'gstadder.c', orc_c, orc_h,
  c_args: gst_plugins_base_args,
  include_directories: [configinc, libsinc],
  dependencies : adder_deps,
  install : true,
  install_dir : plugins_install_dir,
)

plugins += [adder]
```

Centricular