# GPTR Package Structure

## Public API (what users call)

* gpt\_column() – orchestrates per-row prompting + parsing + coercion
* gpt\_chat() – multi-turn chat helper
* gpt() – low-level, single-turn completion
* patch\_failed\_rows() – retry pipeline
* estimate\_tokens() – offline token estimate

## Orchestrators (glue logic)

* gpt\_column\_orchestrator() – calls build\_prompt(), gpt(), json\_fix\_parse\_validate(), finalize\_columns()
* Supports auto\_correct\_keys, keep\_unexpected\_keys, debug capture
* gpt\_chat\_session() – maintains conversation state, calls gpt()

## Engines (do the work)

* gpt() – provider facade
* .detect\_local\_backend() – autodetect backend
* request\_openai(), request\_local() – HTTP POST
* compose\_request\_payload() – messages, files, images
* parse\_response\_text() – extract assistant content

## Parsing & Validation

* json\_fix\_parse\_validate() – repair → parse → validate
* tidy\_json() – flatten nested JSON
* auto\_correct\_keys() – fuzzy match unexpected keys

## Prompting

* build\_prompt(template, text, keys) – injects text & json\_format
* prompt\_presets – curated templates

## Column Finalization

* finalize\_columns() – enforce order, insert missing columns

## Utilities

* estimate\_tokens(), compute\_cost()
* with\_progressr(), update\_eta\_state()
* gptr\_options(), defaults (op.gptr)
* Debug logging
* prepare\_file\_part(), prepare\_image\_part()

## Providers (HTTP clients)

* request\_openai(), request\_local()
* normalize\_http\_error()
* retry\_with\_backoff()

## Testing

* Unit tests – happy path, schema coercion, JSON repairs
* Doubles & fixtures – mock\_gpt(), recorded responses

## Docs & DevX

* Roxygen docs for public functions
* Vignettes
* Articles/snippets
* pkgdown site

## Internal File Layout (suggested)

* R/gpt.R, R/providers\_openai.R, R/providers\_local.R
* R/gpt\_column.R, R/gpt\_column\_orchestrator.R
* R/json\_parse.R, R/prompt.R, R/finalize.R
* R/utils\_progress.R, R/utils\_tokens.R, R/utils\_options.R, R/utils\_http.R
* R/chat.R
* inst/templates/
* tests/testthat/

## Minimal dependency edges

* gpt\_column() → gpt\_column\_orchestrator() → build\_prompt() → gpt() → request\_\*() → parse\_response\_text() → json\_fix\_parse\_validate() → finalize\_columns()
* gpt\_chat() → gpt\_chat\_session() → gpt()