```
create or replace package body ut_file_mapper is
  /*
  utPLSQL - Version 3
  Copyright 2016 - 2021 utPLSQL Project

  Licensed under the Apache License, Version 2.0 (the "License"):
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
  */

  type tt_key_values is table of varchar2(4000) index by varchar2(4000);

  /**
   * Private functions
   */

  function to_hash_table(a_key_value_tab ut_key_value_pairs) return tt_key_values is
    l_result tt_key_values;
  begin
    if a_key_value_tab is not null then
      for i in 1 .. a_key_value_tab.count loop
        l_result(upper(a_key_value_tab(i).key)) := a_key_value_tab(i).value;
      end loop;
    end if;
    return l_result;
  end;

  /**
  * Public functions
  */
  function default_file_to_obj_type_map return ut_key_value_pairs is
  begin
    return ut_key_value_pairs(
        ut_key_value_pair('fnc', 'FUNCTION'),
        ut_key_value_pair('prc', 'PROCEDURE'),
        ut_key_value_pair('tpb', 'TYPE BODY'),
        ut_key_value_pair('pkb', 'PACKAGE BODY'),
        ut_key_value_pair('bdy', 'PACKAGE BODY'),
        ut_key_value_pair('trg', 'TRIGGER')
    );
  end;

  function build_file_mappings(
    a_file_paths                   ut_varchar2_list,
    a_file_to_object_type_mapping ut_key_value_pairs := null,
    a_regex_pattern                varchar2 := null,
    a_object_owner_subexpression   positive := null,
    a_object_name_subexpression    positive := null,
    a_object_type_subexpression    positive := null
  ) return ut_file_mappings is
  begin
    return build_file_mappings(
      null, a_file_paths, a_file_to_object_type_mapping, a_regex_pattern,
      a_object_owner_subexpression, a_object_name_subexpression, a_object_type_subexpression
    );
  end;

  function build_file_mappings(
    a_object_owner                 varchar2,
    a_file_paths                   ut_varchar2_list,
    a_file_to_object_type_mapping ut_key_value_pairs := null,
    a_regex_pattern                varchar2 := null,
    a_object_owner_subexpression   positive := null,
    a_object_name_subexpression    positive := null,
    a_object_type_subexpression    positive := null
  ) return ut_file_mappings is
    l_file_to_object_type_mapping ut_key_value_pairs := coalesce(a_file_to_object_type_mapping,
default_file_to_obj_type_map());
    l_regex_pattern                varchar2(4000) := coalesce(a_regex_pattern, gc_file_mapping_regex);
    l_object_owner_subexpression   positive := coalesce(a_object_owner_subexpression, gc_regex_owner_subexpression);
    l_object_name_subexpression    positive := coalesce(a_object_name_subexpression, gc_regex_name_subexpression);
    l_object_type_subexpression    positive := coalesce(a_object_type_subexpression, gc_regex_type_subexpression);

    l_key_values      tt_key_values;
    l_mappings        ut_file_mappings;
    l_mapping         ut_file_mapping;
```

```
    l_object_type_key varchar2(4000);
    l_object_type     varchar2(4000);
    l_object_owner    varchar2(4000);
    l_file_path       varchar2(32767);
  begin
    if a_file_paths is not null then
      l_key_values := to_hash_table(l_file_to_object_type_mapping);
      l_mappings := ut_file_mappings();

      for i in 1 .. a_file_paths.count loop
        l_file_path := replace(a_file_paths(i),'\','/');
        l_object_type_key := upper(regexp_substr(l_file_path, l_regex_pattern, 1, 1, 'i', l_object_type_subexpression));
        if l_key_values.exists(l_object_type_key) then
          l_object_type := upper(l_key_values(l_object_type_key));
        else
          l_object_type := null;
        end if;

        l_object_owner := coalesce(
          upper(a_object_owner),
          upper(regexp_substr(l_file_path, l_regex_pattern, 1, 1, 'i', l_object_owner_subexpression)),
          sys_context('USERENV', 'CURRENT_SCHEMA'));

        l_mapping := ut_file_mapping(
          file_name    => a_file_paths(i),
          object_owner => l_object_owner,
          object_name  => upper(regexp_substr(l_file_path, l_regex_pattern, 1, 1, 'i', l_object_name_subexpression)),
          object_type  => l_object_type
        );
        l_mappings.extend();
        l_mappings(l_mappings.last) := l_mapping;
      end loop;
    end if;

    return l_mappings;
  end;

end;
/
```