public class CodeSetDelegate {

    private final CodeSetDao dao;

    // Master list

    private List<CodeSetVo> codeSetList = new ArrayList<>();

    // Lookups

    private final Map<String, List<CodeSetVo>> mapByCategory = new HashMap<>();

    private final Map<Integer, CodeSetVo> mapByCode = new HashMap<>();

    public CodeSetDelegate(CodeSetDao dao) {

        this.dao = dao;

    }

    /\*\*

     \* Load all code sets from DB into memory.

     \*/

    public synchronized void loadCodeSet() {

        try {

            this.codeSetList = dao.selectAllRecords();

            mapByCategory.clear();

            mapByCode.clear();

            for (CodeSetVo vo : codeSetList) {

                cache(vo);

            }

            System.out.println("✅ Code sets loaded. Size: " + codeSetList.size());

        } catch (SQLException e) {

            throw new RuntimeException("Error loading code sets", e);

        }

    }

    private void cache(CodeSetVo vo) {

        mapByCategory.computeIfAbsent(vo.getCodeCategory(), k -> new ArrayList<>())

                .add(vo);

        mapByCode.put(vo.getCode(), vo);

    }

    /\*\*

     \* Get all codes for a given category.

     \*/

    public List<CodeSetVo> getCodesByCategory(String category) {

        return new ArrayList<>(mapByCategory.getOrDefault(category, Collections.emptyList()));

    }

    /\*\*

     \* Get description for a given code.

     \*/

    public String getDescriptionByCode(int code) {

        return Optional.ofNullable(mapByCode.get(code))

                .map(CodeSetVo::getDescription)

                .orElse(null);

    }

    /\*\*

     \* Get code for a given category + description.

     \*/

    public int getCodeByCategoryAndDescription(String category, String description) {

        return getCodesByCategory(category).stream()

                .filter(vo -> vo.getDescription().equalsIgnoreCase(description))

                .map(CodeSetVo::getCode)

                .findFirst()

                .orElse(-1);

    }

    /\*\*

     \* Get all descriptions for a category.

     \*/

    public List<String> getDescriptionsByCategory(String category) {

        return getCodesByCategory(category).stream()

                .map(CodeSetVo::getDescription)

                .toList();

    }

    /\*\*

     \* Clear all cache.

     \*/

    public synchronized void clear() {

        codeSetList.clear();

        mapByCategory.clear();

        mapByCode.clear();

    }

}