```python
"""This module implements UserManager utility methods.
"""

# Author: Ling Thio <ling.thio@gmail.com>
# Copyright (c) 2013 Ling Thio

try:
    from urllib.parse import urlsplit, urlunsplit   # Python 3
except ImportError:
    from urlparse import urlsplit, urlunsplit       # Python 2


from flask_login import current_user


# This class mixes into the UserManager class.
# Mixins allow for maintaining code and docs across several files.
class UserManager__Utils(object):
    """Flask-User utility methods."""

    # Flask-Login 0.2 uses functions while 0.3 uses properties
    def call_or_get(self, function_or_property):
        """| Calls ``function_or_property`` if it's a function.
        | Gets ``function_or_property`` otherwise.

        In Flask-Login 0.2 ``is_authenticated`` and ``is_active`` were
        implemented as functions, while in 0.3+ they are implemented as properties.

        Example::

            if self.call_or_get(current_user.is_authenticated):
                pass
        """
        return function_or_property() if callable(function_or_property) else function_or_property

    def email_is_available(self, new_email):
        """Check if ``new_email`` is available.

        | Returns True if ``new_email`` does not exist or belongs to the current user.
        | Return False otherwise.
        """

        user, user_email = self.db_manager.get_user_and_user_email_by_email(new_email)
        return (user == None)

    def generate_token(self, *args):
        """Convenience method that calls self.token_manager.generate_token(\*args)."""
        return self.token_manager.generate_token(*args)

    def hash_password(self, password):
        """Convenience method that calls self.password_manager.hash_password(password)."""
        return self.password_manager.hash_password(password)

    def make_safe_url(self, url):
        """Makes a URL safe by removing optional hostname and port.

        Example:

            | ``make_safe_url('https://hostname:80/path1/path2?q1=v1&q2=v2#fragment')``
            | returns ```'/path1/path2?q1=v1&q2=v2#fragment'``

        Override this method if you need to allow a list of safe hostnames.
        """

        # Split the URL into scheme, netloc, path, query and fragment
        parts = list(urlsplit(url))

        # Clear scheme and netloc and rebuild URL
        parts[0] = ''   # Empty scheme
        parts[1] = ''   # Empty netloc (hostname:port)
        safe_url = urlunsplit(parts)
        return safe_url

    def prepare_domain_translations(self):
        """Set domain_translations for current request context."""
        from .translation_utils import domain_translations
        if domain_translations:
            domain_translations.as_default()
```

```
79
80      def verify_password(self, password, password_hash):
81          """Convenience method that calls self.password_manager.verify_password(password, password_hash).
82          """
83          return self.password_manager.verify_password(password, password_hash)
84
85      def verify_token(self, token, expiration_in_seconds=None):
86          """Convenience method that calls self.token_manager.verify_token(token, expiration_in_seconds)."""
87          return self.token_manager.verify_token(token, expiration_in_seconds)
```