PARDUS APIDOC STANDARDS

Function & Method Documentation

```
def addItem(self, id, name="", mounted=False):
    """This function is used for to add new disk
    unit items to the list.

    Disk unit items are the items that are plugged
    to system. They are
    added to list with related unit information.
    (This information for the developers who wanna
    use this function/method.)

    :param id_: Identification of the disk unit
    :type id_: string
    :param name: Name for this disk unit
    :type name: string
    :param mounted: Current mount state of the unit
    :type mounted: Boolean
    :returns: **True** if operation is succesful,
              otherwise **False**
    :rtype: Boolean
    :raises UnitNotExistException: If id not exists
                                   this occurs

   """

   if mounted:
       result = self.findMounted(id)
       if not result:
           raise UnitNotExistException
       self.list.append(id)
   # This information will be read by the developers
   # who wanna develop or maintain this
   # function/method.
   if self.doSomething():
       return True
   else:
       return False
```

Assume that above code snippet is in a class.

- One line summary (synopsis)
- An extended description (Optional)
- Arguments (If exists)
    - :param arg: `description`
    - :type arg: `type of the above argument`
- Return value (If exists)
    - :returns: `description`
    - :rtype: `type`

Class Documentation

```
class EntryView(QScrollView):
      """ Holds the rule entries and shows them like
      a list view

      This class is a simple listview
      implementation. It behaves like a listview
      when entries are added, deleted or selected.
      The main difference from a listview is this
      class stores custom widget
      elements instead of listviewitems.

      **Item Addition**: add a new item to view
      **Item Deletion**: delete an existing item
                         from view
      **Handling Item Events**: handle item events
   """
```

- One line summary (Synopsis)
- An extended description (Optional)
- Main Jobs (Optional) (*If you believe giving main operation names useful, write them here. For example, a class may have lots of public functions but developer may wanna see most useful public functions and their tasks briefly. (This is not an obligation.)*)
    - **Job Definition**: `Job description`
- Exceptions (If exists)
    - :raisises `Type of the Exception`: `description`

Module Documentation

```
""" This module provides a widget that lists the
disk items with a fancy view...

.. moduleauthor:: pars <admins@pardus.org.tr>
(:author: ?)

"""
```

- One line summary (synopsis)
- Module author
    - .. moduleauthor:: Author Name <author e-mail>

Variable & Attribute Documentation

```
  """ Holds the current index value for a
  ListElement """
  index = 0
```

Assume that below code snippet defined in a class.

- Description
    - What is that attribute, why you defined that?