Conversation API
(Bukkit 1.1-R6 +)

Assumptions

- You are a Bukkit plugin developer
- You have written more then the basic Hello World Plugin.

What is the conversation API?

- A way to easily make structured text menus in SMP.

- Contextual menus.

- Non-linear.

Basic concepts

- Prompts
- ConversationContext
- Conversables
- Conversations

Prompts

- Smallest unit of a conversation.

- Validate text sent to it from a conversable.

- Which prompt to move to.

- Available in many flavours (message, String, Number, Boolean, Regex).

- Extendible

- Exists outside of an actual conversation, so only needs to be built **once**.

Prompt graph

- Collection of interconnected prompts.
- Any types.
- Together they form a potential conversation.
- Non linear paths allowed.

Example prompt graph

Starts at this node
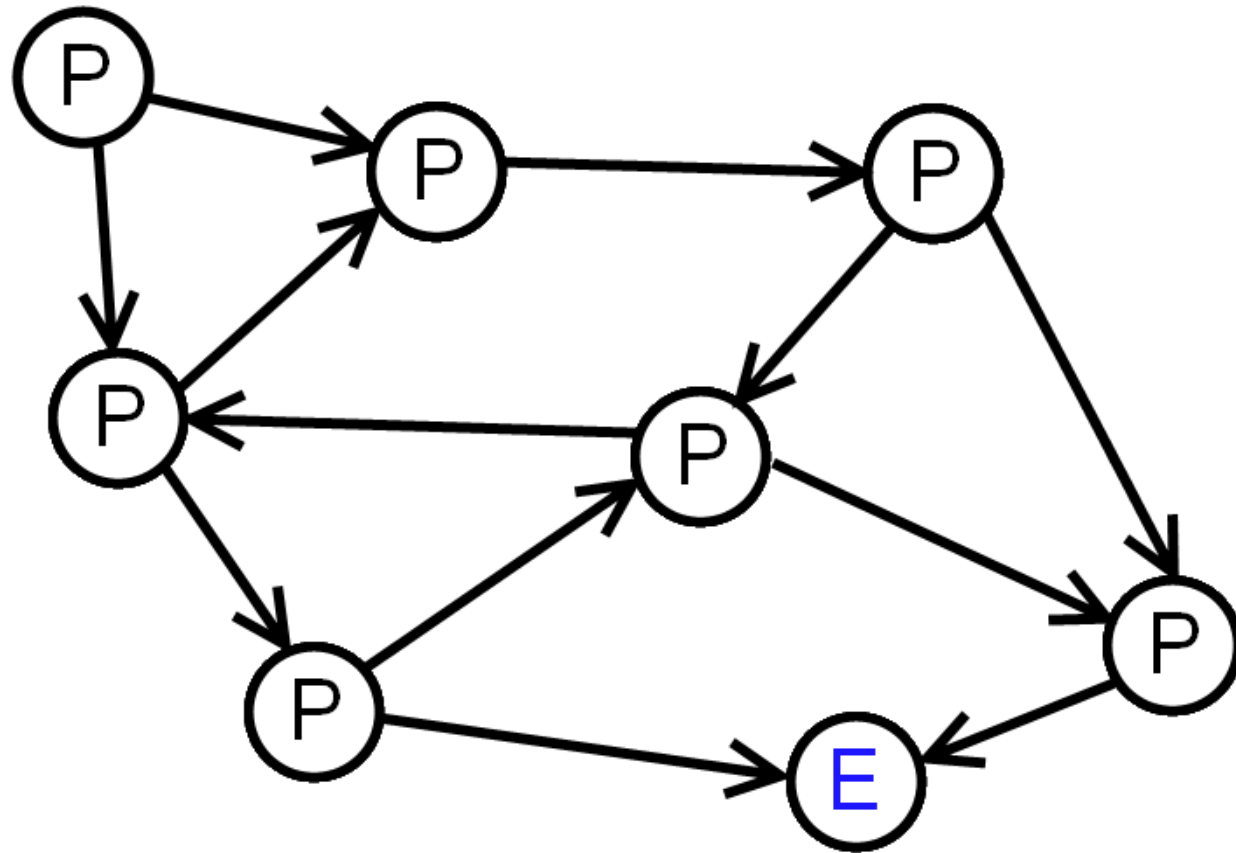
Conversation Context

- Provides prompts with information on user and plugin.

- Provides methods to store/read data related to this instance of conversation.

- Exists only for that conversation, does not carry to new conversations (even of the same type)

Conversables

- An object that can engage in conversation
- Currently only Console and Player

Conversations

- Glue all of the above together.

- Provides prompt graph with the context for it's interaction with a particular player.

Uses of conversations

- Talking to stuff (NPC's)

- Menu Systems

  - Shop menu

  - Knowledge base menu

  - Configuration menu

- Much more... It is a tool, find new ways to use it.

Example - BeardStats

- To look at stats in game, you use the /stats command.

- To find a particular stat takes 3 commands

- /stats -c :: View all categories

- /stats -c blockdestroy :: view all stats in the blockdestroy category

- /stats blockdestroy.grass :: view the stat.

- Must be a better way to do so.

Example – BeardStats Cont.d

- Step in conversations
- Simple 3 prompt graph.
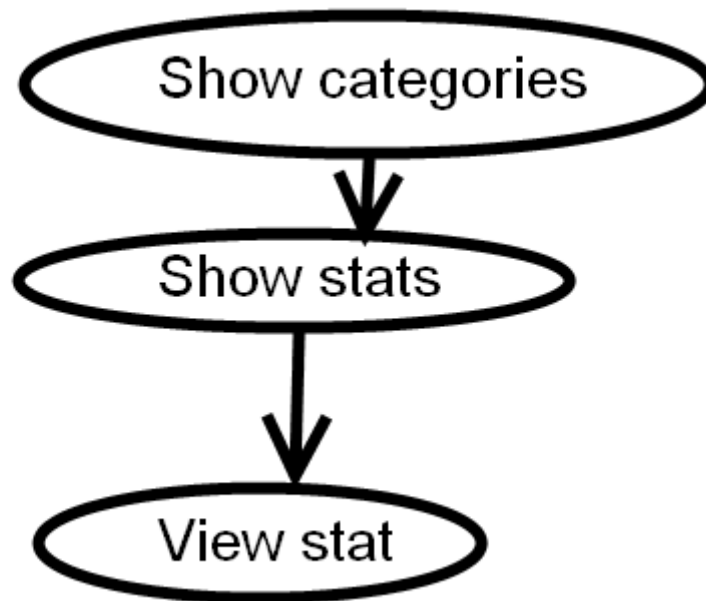
Code to get category from a user

```java
listCategory = new ValidatingPrompt() {

    public String getPromptText(ConversationContext context) {
        CommandSender sender = (CommandSender)context.getForWhom();
        HashSet<String> cats = new HashSet<String>();
        for( PlayerStat ps :playerStatManager.getPlayerBlob(((Player)sender).getName()).getStats()){
            if(!cats.contains(ps.getCat())){
                cats.add(ps.getCat());
            }
        }
        String msg = "";

        Iterator<String> it = cats.iterator();
        while(it.hasNext()){
            for(int i=0;i<10;i++){
                if(it.hasNext()){
                    if(i>0){msg+=", ";}
                    msg+=it.next();
                }
            }
            context.getForWhom().sendRawMessage(ChatColor.AQUA + msg);
            msg="";
        }
        return "Select a category";
    }

    @Override
    protected boolean isInputValid(ConversationContext context, String input) {
        Player sender = (Player)context.getForWhom();

        return playerStatManager.getPlayerBlob(sender.getName()).hasCat(input);
    }

    @Override
    protected Prompt acceptValidatedInput(ConversationContext context,
            String input) {
        context.setSessionData("cat", input);
        return ListStat;
    }
};
```

DO NOT CAST TO PLAYER
AND sendMessage()
IT WILL BE BLOCKED

Set session data for this User for use in the next Prompt.

Methods you override to do so.

- getPromptText()
  - Called when creating the prompt
  - Used in our example to generate a list of categories and ask user to select one.
- isInputValid()
  - Check if what we have is valid.
  - If no, repeat getPromptText()
- acceptValidInput()
  - Return next prompt to move to (in our case, the listStat prompt).

listStat prompt

- Same as listCategory structurally.
- Goes to viewStat.

ViewStat / Accessing session data

```
return   (String) context.getSessionData("cat") +
        "." +
(String)context.getSessionData("stat") +
" : "+
playerStatManager.getPlayerBlob(sender.getName()).getStat(
        (String) context.getSessionData("cat"),
        (String)context.getSessionData("stat")
        ).getValue();
```