

## Functions

Adds a quest to the player by ID.

```
Bool ply:AddQuest( questID )
```

Completes a quest by ID with a status and message to be shown to the user.

```
Void ply:CompleteQuest( questID, status, message )
```

Status Messages:

```
Quest.QUEST_COMPLETE
```

```
Quest.QUEST_ABANDONED
```

```
Quest.QUEST_FAILED
```

Removes a quest from the player. Don't use this, use `CompleteQuest()` instead.

```
Void ply:RemoveQuest( id )
```

Add quest progress to the player.

```
Int ply:AddQuestProgress( questID, num )
```

Get the active quests from the player.

```
Table ply:GetQuests()
```

Get the quests a player has completed.

```
Table ply:GetCompletedQuests()
```

Get if the player has completed the specified quest.

```
Bool ply:HasCompletedQuest( questID )
```

Check if the player is currently on the specified quest.

```
Bool ply:IsOnQuest( questID )
```

Adds the specified amount of money to the player.

```
Int ply:AddMoney( amount )
```

Removes the specified amount of money from the player.

```
Int ply:RemoveMoney( amount )
```

Gets the amount of money the player has.

```
Int ply:GetMoney()
```

Removes money from the player and gives it to another.

```
Void ply:GiveMoney( target, amount )
```

Function to give items to the player with sound effects.

```
Void ply:GiveWeapon( item )
```

Find a quest by ID, returns a table.

```
Table Quest:FindByID( id )
```

Registers a new quest with the quest handler.

```
Void Quest:RegisterQuest( table )
```

Helper function to make hook names. Note that this one is . not :

```
String Quest.HookName( player )
```

Helper function to send a message to a player via hint.

If player isn't specified, it will go to everyone in the server.

```
Void util.Announce( message, [player] )
```

## Quest Structure

Int id

String shortName

String longName

Bool repeatable

Table Data

- Int Start
- Int Target

String desc

Function Accept( player )

Function Abandon( player )

Function Complete( player )

```
Quest:RegisterQuest( table )
```