

# CreateProjectile

This function creates a projectile of the specified type on the specified coordinates.

**Note:**

- **Model** argument is not synchronized between clients. Clients differs from local player see always standard projectile model.
- **Target** argument valid elements are: player, ped, vehicle and object.

## Syntax

```
projectile createProjectile ( element creator, int weaponType [, float posX, float posY, float posZ, float force = 1.0, element target = nil, float rotX, float rotY, float rotZ, float velX, float velY, float velZ, int model ] )
```

**OOP Syntax** Help! I don't understand this!

**Method:** *Projectile(...)*

## Required Arguments

- **creator:** The element representing creator of the projectile. In case you want the projectile to be synced for everybody creator must be the local player or his vehicle.
- **weaponType:** int representing the projectile weaponType (characteristics). Valid IDs are:

ID	Name/Description
16	Grenade
17	Tear Gas Grenade
18	Molotov
19	Rocket (simple)
20	Rocket (heat seeking)
21	Air Bomb
39	Satchel Charge
58	Hydra flare

**Note:** ID 58 doesn't work <sup>[1]</sup>

## Optional Arguments

*NOTE:* When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

- **posX, posY, posZ:** float starting coordinates for the projectile. They are coordinates of creator by default.
- **force:** float representing the starting force for throwable projectiles.
- **target:** element target used for heat seeking rockets.
- **rotX, rotY, rotZ:** float starting rotation for the projectile.
- **velX, velY, velZ:** float starting velocity for the projectile.
- **model:** Integer representing the projectile's model, uses default model for weaponType if not specified.

## Returns

Returns a *projectile* element if projectile creation was successful. Returns *false* if unable to create a projectile (wrong weapon ID or projectiles limit was reached).