

# Superclasse

- **cwbBpaWizard**: Finestra di gestione wizard da estendere

## Interfaccia

- **wizardable**: Interfaccia da implementare

Per creare un wizard va creata una form 'container' con all'interno:

- \* **la buttonBar (tasti gestiti: Avanti, Indietro, Annulla e Concludi).**
- \* **un div segnaposto chiamato 'divGestione' su cui verranno inserite le form dei vari step.**

Questa form container dovrà estendere la classe cwbBpaWizard mentre le form dei singoli step dovranno implementare l'interfaccia wizardable. Sulla form container nel metodo initVars vanno valorizzate le seguenti proprietà:

- \* **nameForm**: il nome della form container
- \* **firstStepName**: Il nome della form utilizzata come primo step
- \* **lastStepName**: Il nome della form utilizzata come ultimo step
- \* **DBName**: Il nome del db da usare per la connessione (es. 'CITYWARE')

Il primo e l'ultimo step sono quindi obbligatoriamente fissi, mentre tutte le form intermedie sono dinamiche.

Nella superclasse è presente una variabile 'navigationRules' che è una linkedlist che gestisce il flusso. La variabile navigationRules ha un metodo current (\$this→navigationRules→current()) che specifica la form corrente in cui ci troviamo.

Nel metodo preNext va gestito il comportamento del wizard al click del tasto avanti (va impostato qual'è lo step successivo e le azioni/parametri da passare) Per aggiungere uno step alla navigationRules va chiamato il metodo addStepToNavigationRules passandogli come parametro il nome della form successiva (\$this→addStepToNavigationRules('nomeDelloStepSuccessivo')).

es implementazione preAvanti:

```
protected function preAvanti($currentStep) {
    switch ($currentStep) {
        case ' cwdForm1':
            $x = $_POST[' cwdForm1_x'];
```

```
switch ($x) {
    case "1":
        $this->addStepToNavigationRules(' cwdForm2'); // se x=1
aggiungo cwdForm2 come step successivo
        // codice....
        break;
    case 2:
        $this->addStepToNavigationRules(' cwdForm3'); // se x=2
aggiungo cwdForm3 come step successivo
        // codice....
        break;
}
break;
case ' cwdForm2':
// se non aggiungo uno step successivo, vado allo step finale
// codice....
break;
}
}
```

Implementando l'interfaccia wizardable, verrà richiesto di implementare il metodo 'validaWizardStep(\$formData, &\$msg)' in cui va gestita la validazione della form al click del tasto 'avanti'. Il metodo deve tornare true o false in base al risultato e concatenare i messaggi di errore su \$msg.

From:  
<https://wiki.nuvolaitalsoft.it/> - wiki



Permanent link:  
[https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:cityware\\_wizard&rev=1467124592](https://wiki.nuvolaitalsoft.it/doku.php?id=sviluppo:cityware_wizard&rev=1467124592)

Last update: **2018/03/19 10:45**