

# **MVC 5**

## **(Model-View-Controller)**

**Rad sa formama,  
POST i GET metode**

# Form element

- Form element koristimo za formiranje obrazaca namijenjenih za unos korisničkih podataka
- Može sadržavati: **input**, **textarea**, **button**, **select**, **option**, **optgroup**, **fieldset** i **label** elemente
- Elemente razlikujemo po njihovom atributu - **name**
- Svi elementi unutar form elementa (njegovi child elementi) biti će poslani kao parametri Request objekta do slijedećeg resursa web aplikacije nakon submita forme

# Atributi form elementa

- Method atribut form elementa specificira način slanja podataka do slijedeće stranice:  
`<form method="get|post">`
- Podaci se mogu slati kao dio URL-a ili kao dio HTTP POST transakcije

# GET metoda

- GET je defaultna metoda
- Nadodaje podatke sadržane u formi (form-data) na URL u obliku parova ime/vrijednost (name/value):  
`URL?name1=value1&name2=value2`
- GET nikada ne koristimo za slanje osjetljivih podataka, za koje nije dobro da su vidljivi u URL-u
- Možemo slati do 3000 znakova na ovaj način
- Ostaje u History-ju preglednika

# POST metoda

- Šalje podatke sadržane u formi (form-data) kao dio tijela HTTP POST transakcije:

`POST /controller/akcija HTTP/1.1`

`Host: www.vern.hr`

`name1=value1&name2=value2`

- Ovo možemo vidjeti uz pomoć alata za debugiranje
- Ovom metodom šaljemo osjetljive podatke
- Nema limita u količini podataka
- Ne ostaje u History-ju preglednika
- Ne skladište (cache) se podaci u memoriji

# Primjer GET metode unutar Controllera

- Mogao bi se staviti [**HttpGet**] atribut na metodu (kao na slijedećoj stranici za POST metodu), ali budući da je GET defaultna opcija onda možemo i bez tog atributa.

```
// GET: Books/Create
```

```
public ActionResult Create()
```

```
{
```

```
    ViewBag.PisacID = new SelectList(db.Writers, "ID", "Ime");
```

```
    ViewBag.ZanrID = new SelectList(db.Genres, "ID", "Naziv");
```

```
    return View();
```

```
}
```

# Primjer POST metode unutar Controllera

```
// POST: Books/Create
```

```
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
```

```
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public ActionResult Create([Bind(Include = "ID,Naslov,PisacID,Kolicina,ZanrID,SlikaURL,Opis,Cijena")] Book book)
{
    if (ModelState.IsValid)
    {
        db.Books.Add(book);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.PisacID = new SelectList(db.Writers, "ID", "Ime", book.PisacID);
    ViewBag.ZanrID = new SelectList(db.Genres, "ID", "Naziv", book.ZanrID);
    return View(book);
}
```