

```

//@route POST api/profile
//@desc Create/Update user's profile
//@access private
router.post(
  '/',
  [auth, [body('status').not().isEmpty().withMessage('Status is required'),
  body('skills').not().isEmpty().withMessage('Skills are required')]],
  async (req, res) => {
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({ errors: errors.array() });
    }
    const {company, website, location, status, skills, bio, githubusername, facebook, linkedin, instagramm}
    = req.body;
    //Build a Profile object
    const profilefields = {};
    profilefields.user = req.user.id;
    if (company) profilefields.company = company;
    if (website) profilefields.website = website;
    if (location) profilefields.location = location;
    if (status) profilefields.status = status;
    if (bio) profilefields.bio = bio;
    if (githubusername) profilefields.githubusername = githubusername;
    if (facebook) profilefields.facebook = facebook;
    if (linkedin) profilefields.linkedin = linkedin;
    if (instagramm) profilefields.instagramm = instagramm;
    if (skills) {profilefields.skills = skills.split(',').map((skill) => skill.trim())}
    //Build a Social object
    profilefields.social = {};
    if (facebook) profilefields.social.facebook = facebook;
    if (linkedin) profilefields.social.linkedin = linkedin;
    if (instagramm) profilefields.social.instagramm = instagramm;
    try {
      let profile = await Profile.findOne({ user: req.user.id });
      if (profile) {
        //UPDATE A PROFILE
        profile = await Profile.findOneAndUpdate({ user: req.user.id }, { $set: profilefields }, { new: true });
        return res.json(profile)
      }
      //CREATE A NEW PROFILE
      profile = new Profile(profilefields);
      await profile.save();
      res.json(profile);
    } catch (error) {
      console.error(error.message);
      res.status(500).send('Server ERROR!')}
  });

```